



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE INGENIERÍA ELECTRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO
ELECTRÓNICO**

PROYECTO TÉCNICO:

**DISEÑO DE ALGORITMOS PARA LA SINCRONIZACIÓN Y
CODIFICACIÓN DE CANALES DE SISTEMAS OFDM
UTILIZANDO RADIOS DEFINIDOS POR SOFTWARE**

AUTORES:

**VÍCTOR ALEJANDRO PINZA CASTRO
JAIME GABRIEL PILAMUNGA MOROCHO**

TUTOR:

ING. CARLOS ALBERTO BOSQUEZ BOSQUEZ, MSc.

GUAYAQUIL – ECUADOR

2019

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.

Nosotros, Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho estudiantes de Ingeniería Electrónica de la Universidad Politécnica Salesiana certificamos que los conceptos desarrollados, análisis realizados y las conclusiones del presente proyecto de titulación son de exclusiva responsabilidad de los autores y es propiedad intelectual de la Universidad Politécnica Salesiana.

Guayaquil, septiembre del 2018

(f)_____

Víctor Alejandro Pinza Castro

C.I. 0923879415

(f)_____

Jaime Gabriel Pilamunga Morocho

C.I. 0924754419

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES.

Nosotros, Víctor Alejandro Pinza Castro con documento de identificación N° 0923879415 y Jaime Gabriel Pilamunga Morocho con documento de identificación N° 0924754419, manifestamos nuestra voluntad de ceder a la Universidad Politécnica Salesiana los derechos patrimoniales en calidad de autores del proyecto de titulación titulado “DISEÑO DE ALGORITMOS PARA LA SINCRONIZACIÓN Y CODIFICACIÓN DE CANALES DE SISTEMAS OFDM UTILIZANDO RADIOS DEFINIDOS POR SOFTWARE”, mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico con mención en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, septiembre del 2018

(f)_____

Víctor Alejandro Pinza Castro

C.I.0923879415

(f)_____

Jaime Gabriel Pilamunga Morocho

C.I. 0924754419

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.

Por medio de la presente doy a conocer que el Proyecto de titulación “DISEÑO DE ALGORITMOS PARA LA SINCRONIZACIÓN Y CODIFICACIÓN DE CANALES DE SISTEMAS OFDM UTILIZANDO RADIOS DEFINIDOS POR SOFTWARE” presentado por el Sr. Víctor Alejandro Pinza Castro y el Sr. Jaime Gabriel Pilamunga Morocho para optar por el título de Ingeniero Electrónico con mención en Telecomunicaciones, se ajusta a las normas establecidas por la Universidad Politécnica Salesiana, por tanto, autorizo su presentación ante las autoridades pertinentes.

Guayaquil, septiembre del 2018

(f)_____

Ing. Carlos Alberto Bosquez Bosquez, MSc.

DIRECTOR DE PROYECTO DE TESIS

DEDICATORIA DE RESPONSABILIDAD.

Nosotros, Víctor Alejandro Pinza Castro con documento de identificación N° 0923879415 y Jaime Gabriel Pilamunga Morocho con documento de identificación N° 0924754419, estudiantes de la Universidad Politécnica Salesiana con sede Guayaquil, declaramos que la responsabilidad de contenido de este proyecto de titulación corresponde exclusivamente y es propiedad intelectual de la Universidad Politécnica Salesiana.

Guayaquil, septiembre del 2018

(f)_____

Víctor Alejandro Pinza Castro

C.I.0923879415

(f)_____

Jaime Gabriel Pilamunga Morocho

C.I. 0924754419

DEDICATORIA.

Todo el esfuerzo y la diligencia aplicada para la terminación y desarrollo de este proyecto, se lo dedico en primer lugar a Dios por guiarme, ayudarme a ser diligente, y darme sabiduría. A mi bella esposa por su apoyo y por ser mi guía e inspiración en todo momento. A mis padres, por su respaldo incondicional a lo largo de mi formación académica.

Víctor Alejandro Pinza Castro

Dedico primeramente mi trabajo a mi Dios. De igual manera, se lo dedico a mi querido padre, Jaime Pilamunga y mi querida madre, Martha Morocho que siempre han estado apoyándome, sino fuera por su apoyo incondicional no estaría donde estoy ahora.

A mi amada esposa Evelyn Fajardo que siempre ha estado junto a mí, brindándome su apoyo día tras día, en las que me ayudo a cumplir esta meta y que gracias a ella puedo lograrlo. A mi familia en general, porque a pesar de tantas circunstancias han estado en los momentos que más los he necesitado. A mi compañero Víctor Pinza, a quien Dios me dio la oportunidad de conocer y ser amigos, logrando superar obstáculos y cumpliendo nuestras metas.

Jaime Gabriel Pilamunga Morocho

AGRADECIMIENTO.

Agradezco a Dios por brindarme en esta vida, el tiempo y el momento para darme la oportunidad de terminar este proyecto, y por brindarme la familia que tengo, quienes me aman y han estado conmigo en el transcurso de este camino hacia mi carrera. A mi amada esposa, quien en estos años estuvo conmigo esperándome el tiempo necesario a que regrese a casa, lo cual me ha dado la inspiración para seguir adelante y no rendirme, cada una de sus palabras y consejos, me enseñaron a no dejar de intentarlo y mejorar, a mis lindas hijas quienes son mi razón de ser.

A mis padres, quienes me han dado su apoyo incondicional, desde siempre, el cual me ha servido de ejemplo e inspiración para poder culminar esta carrera y poder alcanzar esta meta. A mis profesores, por su ejemplo y por su profesionalidad en la contribución a mi formación como ingeniero.

Víctor Alejandro Pinza Castro

Doy gracias a Dios por guiarme llevándome por el camino correcto y mantenerme fuerte a pesar de las adversidades que se presentaron. A mis padres, que me dieron el ejemplo de que todo es posible si te lo propones y luchas por ello. A mi esposa, que durante estos años de carrera ha sabido apoyarme brindándome su amor incondicional.

A mis compañeros, que semestre tras semestre compartimos no solo el aula, sino también conocimientos que hicieron posible el haber podido lograr esta meta. A mis profesores, que no solo nos enseñaron cada materia, sino que se tomaron minutos de sus clases para enseñarnos a ser buenos profesionales. Gracias a todos los que nos brindaron su ayuda en este proyecto.

Jaime Gabriel Pilamunga Morocho

RESUMEN.

El presente proyecto contempla el diseño de algoritmos para la sincronización y codificación de canales de sistemas OFDM utilizando radios definidos *USRP B205mini-i* por software *GNU Radio* como herramienta de *Software*. Se desarrollaron aplicaciones para un banco de pruebas con 6 algoritmos en OFDM.

Se realizó un banco de pruebas para poder simular la recepción, transmisión, de señales, con las cuales podremos analizar y estudiar la ortogonalidad de las señales con la sincronización y codificación de canales en sistemas OFDM. Se analizaron los efectos del ruido, los niveles de recepción, la amplitud en que se transmiten y reciben las señales, las potencias de transmisión, recepción, así como también las representaciones de los diagramas de constelación que representan la relación de amplitud y fase de una portadora modulada digitalmente, para de esta manera práctica se llegue a la comprensión cabal de la modulación de las señales dentro del espectro electromagnético.

En este proyecto presentamos 6 algoritmos en OFDM para la sincronización y la estimación de canal de forma eficiente y robusta mediante el diseño ortogonal de preámbulos tanto en tiempo como en frecuencia.

Se desarrolló un transmisor y un receptor para mostrar la aplicación de los algoritmos, utilizando GNU Radio y el Universal Software Radio Peripheral USRP B205 mini-i.

Para lograr implementar en software la parte de procesamiento de la señal se utilizaron sistemas de lógica programables como FPGA (Field Programmable Gate Array), DSP (Digital Signal Processor) y microprocesadores, ya que con ellos es posible implementar diferentes tipos en el mismo sistema de radio. Realizamos mediante el software radio la modulación/ demodulación, codificación/ decodificación de las señales.

Se realizaron las gráficas en Python, el cual es un lenguaje de programación interpretado, interactivo y orientado a objetos.

ABSTRACT.

The present project contemplates the Design of Algorithms for the Synchronization and Coding of OFDM System Channels Using Defined Radios USRP B205 mini-I by Software GNU Radio as a Software tool. Applications were developed for a test bench with 6 algorithms in OFDM.

A test bench was made to simulate the reception, transmission, signals, with which we can analyze and study the orthogonality of the signals with the synchronization and coding of channels in OFDM systems. The effects of noise, the reception levels, the amplitude in which the signals are transmitted and received, the Powers of transmission, reception, as well as the representation soft he constellation diagrams that represent the amplitude and phase relationship of a digitally modulated carrier, for this practical way you get to the full understanding of the modulation of the signals within the electromagnetic spectrum.

In this project we presented 6 algorithms in OFDM for channel synchronization and estimation in an efficient and robust way by means of the orthogonal design of preambles in both time and frequency.

A transmitter and a receiver were developed to show the application of the algorithms, using GNU Radio and the Universal Software Radio Peripheral USRP B205 mini-i.

In order to implement the signal processing part in software, programmable logic systems such as FPGA (Field Programmable Gate Array), DSP (Digital Signal Processor) and Microprocessors were used to implement different types in the same radio system. We performed the modulation / demodulation, coding / decoding of the signal using the radio software.

Charts were made in Python, which is an interpreter, interactive and object-oriented programming language.

ÍNDICE GENERAL.

PORTADA.....	I
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.	II
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES.	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.	IV
DEDICATORIA DE RESPONSABILIDAD.	V
DEDICATORIA.	VI
AGRADECIMIENTO.	VII
RESUMEN.	VIII
ABSTRACT.....	IX
ÍNDICE GENERAL.	X
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE TABLAS.	XIX
INTRODUCCIÓN.	1
1. EL PROBLEMA.....	2
1.1. PLANTEAMIENTO DEL PROBLEMA.	2
1.2. ANTECEDENTES.	3
1.3. IMPORTANCIA DEL PROYECTO.....	3
1.4. ALCANCE DEL PROYECTO.....	4
1.5. DELIMITACIÓN DEL PROBLEMA.....	4

1.6. EXPLICACIÓN DEL PROBLEMA.	4
1.7. OBJETIVOS DE LA INVESTIGACIÓN.	4
1.7.1. OBJETIVO GENERAL.	4
1.7.2. OBJETIVOS ESPECÍFICOS.	5
2. FUNDAMENTOS TEÓRICOS.	5
2.1. SISTEMA DE COMUNICACIONES.	5
2.2. SISTEMAS DE COMUNICACIONES DIGITALES.	6
2.2.2. VENTAJAS DE UN SISTEMA DE COMUNICACIONES DIGITALES.	8
2.2.3. MULTIPLEXACIÓN POR DIVISIÓN ORTOGONAL EN FRECUENCIA (OFDM).	10
2.2.3.1. SISTEMA GENERAL DE OFDM.	10
2.2.3.2. USO DE LA FFT EN OFDM.	11
2.2.4. TRANSMISOR Y RECEPTOR OFDM.	12
2.2.4.1. TRANSMISOR OFDM.	12
2.2.4.2. RECEPTOR OFDM.	13
2.2.5. TIEMPO DE GUARDA Y PREFIJO CÍCLICO.	13
2.2.6. MODELO DE SISTEMA DE TIEMPO DISCRETO OFDM.	14
2.2.7. DEFICIENCIAS DEL SISTEMA OFDM.	16
2.2.8. VENTAJAS DE OFDM.	17
3. MARCO METODOLÓGICO.	18

3.1. TIPO DE INVESTIGACIÓN.	18
3.2. DISEÑO DE INVESTIGACIÓN.	18
3.3. ENFOQUE DE LA INVESTIGACIÓN.	18
3.4. METODOLOGÍA DE LA INVESTIGACIÓN.	19
3.5. INSTRUMENTOS DE INVESTIGACIÓN.	19
3.5.1. GNU RADIO.	19
3.5.2. GNU RADIO-COMPANION.....	22
3.5.3. PYTHON.	23
3.5.4. RADIO DEFINIDO POR SOFTWARE.....	23
3.5.4.1. ESTRUCTURA.	24
3.5.4.2. UNIVERSAL SOFTWARE RADIO PERIPHERAL (Sistema USRP).	25
3.5.4.3. CARACTERÍSTICAS.	25
3.5.4.4. APLICACIONES.....	27
3.5.5. CARACTERÍSTICAS GENERALES DEL HARDWARE PROPUESTO EN LA SOLUCIÓN.....	27
3.5.6. UBUNTU.....	29
4. PROPUESTA.....	31
4.1. TÍTULO DE LA PROPUESTA.....	31
4.2. JUSTIFICACIÓN.	31
4.3. DESCRIPCIÓN DE LA PROPUESTA.....	31

5. RESULTADOS.....	34
5.1 ANÁLISIS DE RESULTADOS.....	34
6. CONCLUSIONES.	39
7. RECOMENDACIONES.....	39
8. PRESUPUESTO.....	39
9. CRONOGRAMA.....	40
REFERENCIAS BIBLIOGRÁFICAS.....	41
GLOSARIO.	42
ANEXOS.	47
GUÍAS DE PRÁCTICAS DE LABORATORIO.....	47
PRÁCTICA 1.....	47
PRÁCTICA 2.....	60
PRÁCTICA 3.....	63
PRÁCTICA 4.....	66
PRÁCTICA 5.....	77
PRÁCTICA 6.....	81
PRÁCTICA 7.....	94
PRÁCTICA 8.....	120
ACTA DE ENTREGA – RECEPCIÓN DE TESIS.	132

ÍNDICE DE FIGURAS.

Figura 1 Diseño de un sistema de radio receptor tradicional.....	1
Figura 2 Diagrama de un sistema de comunicación	6
Figura 3 Elementos principales de un sistema de comunicación.....	6
Figura 4 Multiplexor-demultiplexor por división de tiempo	7
Figura 5 (a) Transmisión y conmutación no integradas; (b) Integración de la transmisión y conmutación	9
Figura 6 Comparación entre Single Carrier, Multicarrier y OFDM	10
Figura 7 Modelo de un sistema OFDM simple.....	11
Figura 8 Intervalo de guarda y prefijo cíclico.....	13
Figura 9 Estructura de un símbolo OFDM	14
Figura 10 El principio básico de la propagación multitrayecto.	15
Figura 11 Diagrama de bloques de un Receptor OFDM.	17
Figura 12 Componentes del esquema SDR	20
Figura 13 Niveles de abstracción GNU Radio.....	20
Figura 14 Interfaz de trabajo GNU Radio-companion	22
Figura 15 Diagrama de bloques de SDR.....	24
Figura 16 Sistema de Radio Definido por Software	25
Figura 17 Diagrama de bloques de la estructura interna de un USRP.....	26
Figura 18 USRP B205mini-i.....	27
Figura 19 Arquitectura USRP B205 mini-i	28
Figura 20 Mapa conceptual sobre comunicaciones inalámbricas.....	32
Figura 21 Arquitectura USRP B200mini Series.	32
Figura 22 Pantalla Principal de Oracle VM VirtualBox	48

Figura 23 Crear máquina virtual.	49
Figura 24 Tamaño de la memoria RAM de la máquina virtual.	49
Figura 25 Creación del disco duro virtual.....	50
Figura 26 Tipo de archivo de disco duro	50
Figura 27 Tamaño del disco duro.	51
Figura 28 Ubicación del archivo y tamaño.	51
Figura 29 Propiedades de la máquina virtual.....	52
Figura 30 Agregar dispositivo.	52
Figura 31 Cargar imagen del Ubuntu.....	53
Figura 32 Instalar Ubuntu.	53
Figura 33 Tipo de instalación.	53
Figura 34 Configuración de zona horaria.	54
Figura 35 Configuración del teclado.....	54
Figura 36 Asignar nombre y contraseña a la máquina virtual.	55
Figura 37 Escritorio de Ubuntu.....	55
Figura 38 Instalación del driver del USRP en Windows	56
Figura 39 Instalación USB driver	56
Figura 40 Abriendo USB driver.....	57
Figura 41 USRP B205mini-i.....	58
Figura 42 Descarga de Software	64
Figura 43 Archivos previas a la ejecución de la aplicación.....	65
Figura 44 GNU Radio	67
Figura 45 GNU Radio: bloques	68
Figura 46 Bloque Propiedades	68

Figura 47 Bloque Variables	69
Figura 48 Bloque Signal Source	69
Figura 49 Funciones de generador de señales.....	70
Figura 50 Bloque Noise source.....	70
Figura 51 Bloque Add.....	71
Figura 52 Propiedades UHD: USRP Source.....	72
Figura 53 Propiedades UHD	72
Figura 54 Bloque Throttle.....	73
Figura 55 Bloque Channel Mode	73
Figura 56 Bloque Multiply Const	74
Figura 57 Bloque OFDM Mod	74
Figura 58 Bloque QT GUI Sink.....	75
Figura 59 Bloque Interleaver	75
Figura 60 Ejecución de aplicación.....	76
Figura 61 Instalación de Qt 4.8.6.....	78
Figura 62 Ejecución de aplicación.....	79
Figura 63 Calibración de parámetros	80
Figura 64 Campo de preámbulo.....	82
Figura 65 Creación de GRC.....	83
Figura 66 Proceso de creación de GRC	83
Figura 67 Rangos QT GUI.....	84
Figura 68 Parámetros del bloque opciones	84
Figuras 69 y 70 Creación de variables fft_len y cp_len respectivamente..	85
Figuras 71 y 72 Variables SNR y samp_rate respectivamente	86

Figura 73 Bloque QT GUI Entry	87
Figura 74 Bloque Random Source	87
Figura 75 Bloque Stream to Tagged Stream.....	88
Figura 76 Bloque jerárquico para la modulación OFDM.	89
Figura 77 Bloque Rational Resampler	89
Figura 78 Propiedades de bloque Rational Resampler	90
Figura 79 Bloque OFDM Synchronizer.....	90
Figura 80 Bloque de estimación para señales OFDM	91
Figura 81 Parámetros del bloque UHD: USRP Source	91
Figura 82 Propiedades del bloque UHD: USRP Source	91
Figuras 83, 84, 85 y 86 Aplicación en ejecución.....	93
Figuras 87, 88, 89 y 90 Ingreso de parámetros de transmisión-recepción de una señal en el sistema OFDM	96
Figuras 91 y 92 Parámetros de transmisión-recepción de señal en el sistema OFDM.....	96
Figuras 93 y 94 Parámetros bloque Options	97
Figuras 95 Bloque Import	98
Figuras 96, 97, 98, 99 Bloque Import: Creación de variables	99
Figura 100 Ingreso de sync_word1	100
Figura 101 Ingreso de sync_word2.....	101
Figura 102 Ingreso de occupied_carriers.....	102
Figura 103 Ingreso de pilot_carriers	102
Figura 104 Valor del value	103
Figura 105 Propiedades WX GUI: Label ganancia	104

Figura 106	Propiedades WX GUI: Label frecuencia del muestreo	104
Figura 107	Parámetros UHD, USRP Source: General.....	105
Figura 108	Parámetros UHD, USRP Source: Opciones RF	105
Figura 109	Propiedades WX GUI FFT Sink.....	106
Figura 110	Taps iniciales del ecualizador.....	106
Figura 111	Bloque Delay	107
Figura 112	Bloque modulador de la frecuencia	108
Figura 113	Bloque WX GUI Slider	109
Figura 114	Bloque Multiply.....	109
Figura 115	Bloque WX GUI Constellation Sink	110
Figura 116	WX GUI Waterfall Sink.....	110
Figura 117	Bloque Header/Payload Demux	111
Figura 118	Bloque Constellation Decoder.....	112
Figura 119	Bloque FFT.....	112
Figura 120	Propiedades canal OFDM.....	113
Figura 121	Corrección de frecuencia.....	114
Figura 122	Serializador OFDM	115
Figura 123	Aplicación ejecutándose	116
Figuras 124, 125, 126	Recepción de Señal OFDM	117
Figuras 127, 128, 129	Recepción de Señal OFDM: Proceso.....	118
Figura 130	Recepción de Señal OFDM: Frecuencia de muestreo	119
Figura 131	Creación de grc de receptor WiFi.....	122
Figura 132	Grc de receptor WiFi.....	123
Figura 133	Bloque QT GUI Chooser.....	123

Figura 134 Parámetros de bloque QT GUI Chooser.....	124
Figura 135 Parámetros de bloque QT GUI Chooser: Valor de options ...	124
Figura 136 Parámetros de bloque QT GUI Chooser: Valor de labels	125
Figura 137 Bloque QT GUI Range	126
Figura 138 Bloque UHD USRP Source: parámetros	126
Figura 139 Bloque WiFi Sync Short.....	127
Figura 140 Bloque WiFi Sync Long	127
Figura 141 Sincronización larga Wifi.....	128
Figura 142 Ecualizador de frame de WiFi.....	128
Figura 143 Decodificador WiFi MAC	129
Figura 144 Conector Wireshark.....	129
Figura 145 Bloque File Sink	130
Figura 146 y 147 Paso 3: Ejecución de aplicación	131

ÍNDICE DE TABLAS.

Tabla No. 1 Módulos GNU Radio	21
Tabla No. 2 Estructura de un módulo GNU Radio	22
Tabla No. 3 Tabla de comandos	31
Tabla No. 4 Presupuesto para el proyecto de titulación.....	39
Tabla No. 5 Cronograma.....	40

INTRODUCCIÓN.

Software Defined Radio (SDR).

El concepto de Radio Definido por software o SDR fue concebido en la década de los años ochenta y su propósito inicial fue la creación de dispositivos capaces de emular sistemas de radio operando en diferentes frecuencias o para diversas aplicaciones, inicialmente con fines militares.

Reducir la utilización de hardware especializado y dedicado exclusivamente para el procesamiento de las señales es posible gracias al diseño de funciones y algoritmos que puedan ser ejecutados desde cualquier computadora de propósito general. En este paradigma, la señal debe ser generada digitalmente y manipulada desde la computadora tanto como sea posible. La implementación de moduladores, filtros, bloques de FFT (Fast Fourier Transform), amplificadores, etc., configurados mediante software permiten que la señal esté lista para ser enviada al aire.

Una vez tratada la señal, el USRP se encarga de transformar las muestras digitales en una señal analógica, se modula la señal en banda base con la portadora en la frecuencia deseada para ser transmitida. El software transfiere los datos al hardware, que tiene la función de transformar las muestras digitales en una señal analógica y la modulación de señal de la banda de base a la frecuencia portadora deseada para ser transmitidos. La última etapa sería el envío a la antena.

El concepto de SDR es muy diferente a las radios tradicionales que se han estado utilizando hasta ahora. Estos se basan en hardware dedicado para todas sus funciones y cada parte de hardware tiene una función muy concreta y fija, como lo podemos ver en la siguiente ilustración.

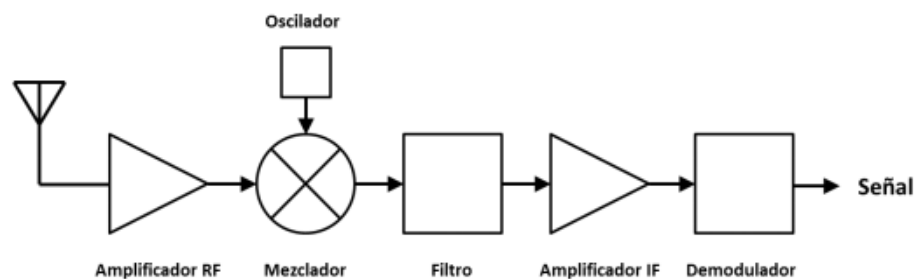


Figura 1 Diseño de un sistema de radio receptor tradicional

Fuente: Víctor Alejandro Pinza Castro - Jaime Gabriel Pilamunga Morocho

Prescindir de hardware dedicado ofrece una gran ventaja respecto al diseño tradicional y esta es una de las principales razones de la utilización de SDR en el área académica y de investigación, ya que nos permite probar diferentes configuraciones tan solo modificando parámetros de operación en el software. Esto ayuda al desarrollo de aplicaciones novedosas mucho más fáciles, rápidas y económicas.

En este proyecto de titulación se realiza un banco de pruebas con el diseño de 6 algoritmos para poder simular la recepción y transmisión de señales, con las cuales se analiza y estudia la ortogonalidad de la señal con la sincronización y codificación de canales en sistemas OFDM mediante el software GNU Radio y hardware USRP B205 mini-i. Se analizan los efectos del ruido, los niveles de recepción, así como la amplitud en que se transmiten y reciben las señales, para que de manera práctica se logre la compresión de la modulación de señales dentro del espectro electromagnético.

El presente proyecto implementa y establece diversos tipos de bancos de prueba para comprobar velocidades de transmisión de varias tecnologías OFDM, donde se puede estudiar las separaciones de las portadoras, modulación y demodulación de las señales, la forma en que viajan las señales OFDM, y como las aprovechan diferentes tecnologías, el proyecto está dirigido para ser usado en el laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana Sede Guayaquil, con el cual se tendrá un conocimiento práctico para así demostrar todas las ventajas que la técnica OFDM ofrece frente a los efectos de los multi-caminos en un entorno de laboratorio para transmisiones inalámbricas, interferencia inter-portadora, entre otras.

1. EL PROBLEMA.

1.1. PLANTEAMIENTO DEL PROBLEMA.

En la actualidad los avances tecnológicos en comunicación inalámbrica han evolucionado a pasos agigantados, tal que existen diferentes tipos de señales que pueden ser utilizadas como medio de transmisión de información. Cada uno de esos sistemas opera con diferentes tipos de configuraciones y modulaciones que hacen posible manipular la señal de transmisión. Entre estas modulaciones está el OFDM (del inglés Orthogonal Frequency-Division Multiple Access) la cual, es una técnica de modulación muy utilizada hoy en día en la mayoría de tecnologías inalámbricas existentes por ejemplo LTE, WIFI, TDT, GSM, etc. Por tal motivo, el estudio de ésta debe reforzarse dentro de las materias de la carrera de Ingeniería Electrónica mencionada en Telecomunicaciones.

Sin embargo, existe una carencia de las prácticas donde se analicen y estudien las señales obtenidas tras utilizar las técnicas de OFDM, así como también de la codificación de canales que busca añadir suficiente redundancia y protección para que las señales corrijan errores. Esto a nivel de laboratorio para reforzar la teoría de comunicaciones inalámbricas, y que el estudiante llegue a tener la comprensión de cómo funcionan estas tecnologías, mediante su propagación y modulación.

1.2. ANTECEDENTES.

El diseño e implementación de algoritmos para la sincronización y codificación de canales de sistemas OFDM utilizando radios definidos por Software, consiste en elaborar e implementar aplicativos realizados en GNU Radio para practicar y utilizar módulos USRP, en cuanto a varias tecnologías que usan OFDM, tales como LTE, GSM, WIFI, TDT. El presente proyecto demuestra seis algoritmos de estudio para la simulación de recepción y transmisión de señales, mediante la sincronización y codificación de canales con un sistema OFDM, mediante la utilización de un software GNU Radio y la implementación de un dispositivo USRP B205 mini-i, como elemento que cumple las dos funciones. Estos materiales permiten al proyecto ser mejorado a través de futuros estudios de señales y sistemas OFDM, que permitan profundizar en los conocimientos de esta técnica.

OFDM es una técnica de modulación muy utilizada hoy en día en la mayoría de las tecnologías inalámbricas existentes, por ejemplo, LTE, WIFI, TDT, GSM, etc. Por tal motivo su estudio se debe reforzar dentro de las materias de la carrera de Ingeniería Electrónica mención en Telecomunicaciones, sin embargo, se carecen de prácticas donde se analicen y estudien las señales obtenidas al utilizar las técnicas de OFDM, así como también de la codificación de canales que buscan añadir suficiente redundancia y protección de la señal para hacer sea más fuerte, en términos de corregir el error. La práctica de laboratorio es muy útil para reforzar la teoría de comunicaciones inalámbricas, permitiendo al estudiante a comprender cómo funcionan estas tecnologías, mediante su propagación y modulación.

Mediante los conceptos básicos de las telecomunicaciones, los cuales son de transmitir y de recibir información de un lugar a otro, cualquiera que sea el tipo de información o datos, siempre se realizará desde un transmisor hasta un receptor. Tomado esto en cuenta como base de este proyecto, se desea transmitir datos de portadoras OFDM y recibir datos de otras técnicas de comunicación digital para cumplir con el objetivo de crear bancos de pruebas aplicados en los diferentes tipos de tecnología OFDM.

1.3. IMPORTANCIA DEL PROYECTO.

En este proyecto de titulación se propone realizar un banco de pruebas con el diseño de 6 algoritmos para poder simular la recepción, transmisión, de señales, con las cuales podremos analizar y estudiar la ortogonalidad de las señales con la sincronización y codificación de canales en sistemas OFDM mediante el software GNU Radio y hardware USRP B205 mini-i. Se analizarán los efectos del ruido, los niveles de recepción, la amplitud en que se transmiten y reciben las señales, para de esta manera práctica se llegue a la comprensión cabal de la modulación de las señales dentro del espectro electromagnético.

1.4. ALCANCE DEL PROYECTO.

Gracias a la implementación de esta propuesta, los futuros estudiantes de la carrera de Ingeniería en Electrónica, pueden poner en práctica los conocimientos adquiridos en el aula; podrán visualizar los gráficos que demuestran las diferencias en la señal inalámbrica, cuando en esta se da una interferencia, cuando se modula la frecuencia, el canal de transmisión, etc. Así, se puede afianzar mejor los conocimientos adquiridos y se da la oportunidad de un estudio más profundo de estos medios de comunicación, los cuales siguen evolucionando constantemente.

La realización de este proyecto refuerza los conocimientos adquiridos en el estudio de las nuevas tecnologías y métodos de modulación que necesita poseer el futuro profesional de la carrera de Telecomunicaciones y Electrónica, con la finalidad de ampliar sus conocimientos y mejorar el desarrollo de sus habilidades como estudiante de la carrera.

1.5. DELIMITACIÓN DEL PROBLEMA.

Con este proyecto se desea implementar y establecer diferentes tipos de bancos de prueba para comprobar velocidades de transmisión de varias tecnologías OFDM, donde se podrán estudiar las separaciones de las portadoras, la modulación y demodulación de las señales, la forma en que viajan las señales en OFDM, y como la aprovechan diferentes tecnologías, todo esto se realizará en el laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana Sede Guayaquil, con los cuales se tendrá un conocimiento práctico para demostrar todas las ventajas que la técnica OFDM ofrece en un entorno de laboratorio para transmisiones inalámbricas, interferencia inter- portadora (Ortogonalidad), entre otras. En sí, el proyecto, a través de las prácticas, demuestra un trabajo de modulación, sincronización y codificación de canales en sistemas OFDM.

1.6. EXPLICACIÓN DEL PROBLEMA.

-Cómo se validan algoritmos para la sincronización y codificación de canales de sistemas y tecnologías OFDM mediante USRP B205 mini i y GNU Radio?

-Por qué es importante desarrollar aplicaciones para un receptor IEEE 802.11a/b/g/p?

-Para qué diseñar aplicaciones mediante codificación de canales?

1.7. OBJETIVOS DE LA INVESTIGACIÓN.

1.7.1. OBJETIVO GENERAL.

Diseñar, simular y validar algoritmos para un banco de prueba para la sincronización y codificación de canales de sistemas y tecnologías OFDM

mediante USRP B205 mini i y GNU Radio implementado en el laboratorio de Telecomunicaciones de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana Sede Guayaquil.

1.7.2. OBJETIVOS ESPECÍFICOS.

- Desarrollar seis algoritmos para un banco de pruebas en OFDM utilizando GNU Radio.
- Simular e implementar en un SDR, el Espectro OFDM, la recepción y transmisión de paquetes OFDM, la modulación y demodulación de OFDM en las tecnologías más conocidas utilizando módulos USRP B205 mini-i para la transmisión y recepción de las señales.
- Diseñar y programar aplicaciones para un receptor IEEE 802.11a/b/g/p, en el cual se realizará el análisis del tráfico que viaja por estas señales, así como también la forma en que se propaga la misma.
- Diseñar aplicaciones mediante codificación de canales para analizar su rendimiento en términos de tasa de error de bits (BER), para el mejoramiento en la eficiencia espectral.

2. FUNDAMENTOS TEÓRICOS.

2.1. SISTEMA DE COMUNICACIONES.

A lo largo del tiempo el deseo del hombre ha sido, poder transmitir información a otras personas, a pesar de la distancia que los separe; esto lo motiva a descubrir nuevas alternativas de comunicación. Esto lo lleva al diseño de complejos sistemas que le permiten cumplir con su objetivo; por ello un sistema de comunicaciones tiene como objetivo principal transportar información de un lugar a otro a través de procesos definidos. (López, Zinemanas, & Ramos, 2016). Dentro del sistema de comunicación, existen tres procesos por los cuales la información debe atravesar; transmisión, procesamiento y recepción, donde los circuitos eléctricos y electrónicos son la fuente principal para que se cumpla dicho proceso. (Carlson, Crilly, & Rutledge, 2015).

Autores como (Fernández Díaz, 2015) indica que un sistema de comunicación puede ser de dos tipos básicos: el analógico y el digital. El primero, y más utilizado desde que se pueden emitir mensajes a largas distancias, el analógico, consiste en la transmisión de energía electromagnética a través de una señal de onda senoidal. Mientras que la segunda, actualmente utilizada para la mayoría de transmisiones de información, la digital, es aquel sistema que transmite y recibe la energía electromagnética a través de niveles discretos como +5v y tierra.

La fuente de información de un sistema de comunicación, sea este analógico o digital, siempre tiene que ser convertida antes de ser propagada a través del canal de transmisión; depende del uso o aplicación de la misma, que se debe configurar cada sistema, siendo los medios de transmisión, los que

sufren los cambios más significativos con el pasar del tiempo.

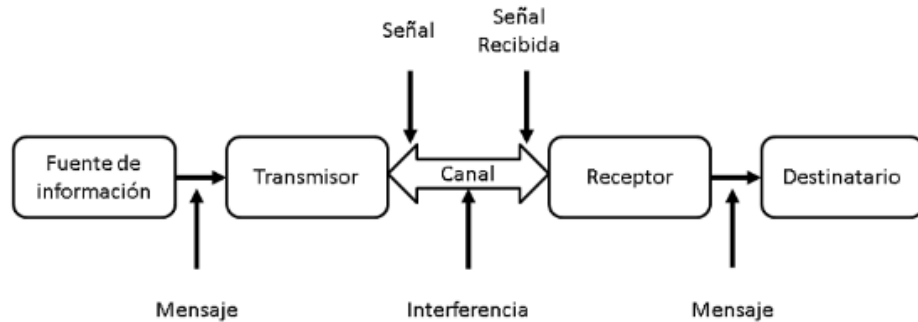


Figura 2 Diagrama de un sistema de comunicación
Fuente: (López, Zinemanas, & Ramos, 2016)

Dentro del proceso de transmisión de información, desde su origen hasta su destino, se cuentan con diversos elementos que son necesarios para que la condición se cumpla; en primer lugar, la fuente o emisor de la información; segundo, el transmisor o conjunto de dispositivos eléctricos y circuitos electrónicos que acogen la fuente de información para convertirla al medio o canal de transmisión; tercero, canal de transmisión, compuesto por los diferentes elementos que hacen posible transportar el mensaje; cuarto, receptor o dispositivo de decodificación de la señal del canal de transmisión; y quinto, destino de la información. (López, Zinemanas, & Ramos, 2016).

2.2. SISTEMAS DE COMUNICACIONES DIGITALES.

Los sistemas de comunicación digital (DCS) se han convertido rápidamente en los más usados para transmitir y recibir información, porque a diferencia de los sistemas de comunicación analógicos, el tratamiento y procesamiento de las señales es mucho más fácil y a diferencia de un sistema analógico, un emisor solo puede tener un conjunto finito de formas de onda discretas que deben coincidir con los bits a transmitir. En este momento, el número de formas de onda que se pueden transmitir es infinito; todo esto de manera independiente al tipo de información que se desee transmitir. “En los sistemas de comunicación digital, los mensajes pertenecen a un conjunto de valores finitos y discretos y son menos sensibles a la superposición de ruido durante la transmisión.”. (Suárez, 2015).

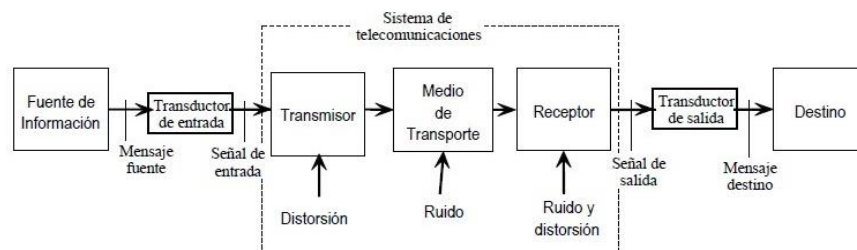


Figura 3 Elementos principales de un sistema de comunicación
Fuente: (Viveros, 2015)

Los sistemas de comunicaciones digitales están conformados básicamente por un transmisor, un canal de transmisión y un receptor; en ellos, la señal de información que se desea transmitir, generalmente está en forma analógica, aunque también existen sistemas en donde la señal de información se encuentra en forma digital. (Quesada, 2016). Si la señal de información es analógica, es convertida a digital a través de pulsos digitales antes de ser transmitida y debe ser reconvertida a analógica en el receptor.

Estos sistemas de comunicación digital permiten al hombre evolucionar su forma de comunicarse con otros; una de las virtudes de los DCS es la producción de una tasa de error binario (BER) relativamente baja (Suárez, 2015). El BER “El número de bits recibidos con el número de errores dividido por el número total de bits recibidos durante el tiempo dado” (Andreotti, 2015); básicamente es la relación existente entre la cantidad de bits recibidos erróneamente y el total de bits recibidos durante un intervalo de tiempo. Entre más baja sea la tasa de error binario, la calidad de transmisión es alta; en cambio si el BER es alto, la calidad de transmisión decae, en cuyo caso dicha cantidad de bits errados, deben ser retransmitidos por el sistema.

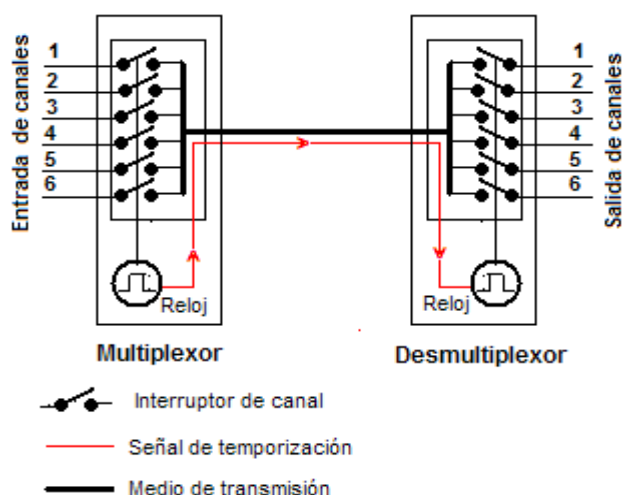


Figura 4 Multiplexor-demultiplexor por división de tiempo

Fuente: (Muñoz, 2018)

En la actualidad la comunicación digital permite a los usuarios mantenerse en contacto con otras personas de manera instantánea, pero esto solo se logra a través de un sistema de comunicación digital eficiente, donde el tratamiento y procesamiento de la información es aprovechada en las diferentes señales utilizadas. La técnica más común para los sistemas digitales es la multiplexación por división de tiempo (TDM), la cual “permite la transmisión de señales digitales y su idea es ocupar el canal (generalmente de gran capacidad) de transmisión desde otra fuente, y de esta manera se logra un mejor uso del medio de transmisión.”. (Muñoz, 2018)

2.2.2. VENTAJAS DE UN SISTEMA DE COMUNICACIONES DIGITALES.

Un sistema de comunicación digital a diferencia de uno analógico, presenta muchas ventajas por las cuales su utilización es la mejor elección para transmitir información.

Entre estas destacan las siguientes, según (Suárez, 2015) y (Muñoz, 2018):

Facilidad para diversos servicios: La digitalización de la información permite transmitirla de diferentes maneras a través de un sistema de comunicación digital, así pues, en un mismo canal de transmisión puede viajar imagen, audio, video, etc., permitiendo que la interacción hombre – tecnología sea más dinámica.

Facilidad de multiplexación: La compartición del mismo medio de transmisión en varias señales, permite aprovechar al máximo el medio donde viaja la información; lo cual ha permitido que la tecnología avance a pasos agigantados en los últimos años.

Facilidad para la utilización de tecnología moderna: Los sistemas de comunicación digital, a diferencia del analógico, permiten un mejor aprovechamiento de los avances tecnológicos, más en concreto a la utilización de tecnología electrónica; “por lo tanto, resulta claro que los desarrollos de la tecnología de los circuitos integrados digitales para circuitos lógicos de computación y memoria se aplican directamente en los equipos de conmutación y transmisión digital”. (Viveros, 2015, pág. 15).

Regeneración de la señal: En la transmisión digital de información, el conjunto finito de datos, también llamados paquetes, permite operar con una mayor facilidad que un sistema analógico, porque esa limitación de información por cada paquete de datos, es lo que evita errores en la transmisión de los mismos.

Así, se pueden “...colocar receptores intermedios que reconozcan los estados antes que la señal se degrade en demasía. Una vez realizado ese reconocimiento, podrá transmitirse el mensaje nuevamente, libre de contaminaciones y con un nivel adecuado para que llegue hasta otro receptor” (Viveros, 2015, pág. 18). Estos receptores intermedios en conjunto con sus transmisores asociados reciben el nombre de repetidores regenerativos o simplemente regeneradores.

Integración de la transmisión y la conmutación: Dentro de los avances tecnológicos que se han realiza a lo largo de los últimos años, se tiene que la unión de dos sistemas que se consideraban como independientes, estos son la multiplexación por división de tiempo y los equipos de conmutación digital.

En la siguiente ilustración, se puede apreciar la simplicidad de la integración de ambas tecnologías:

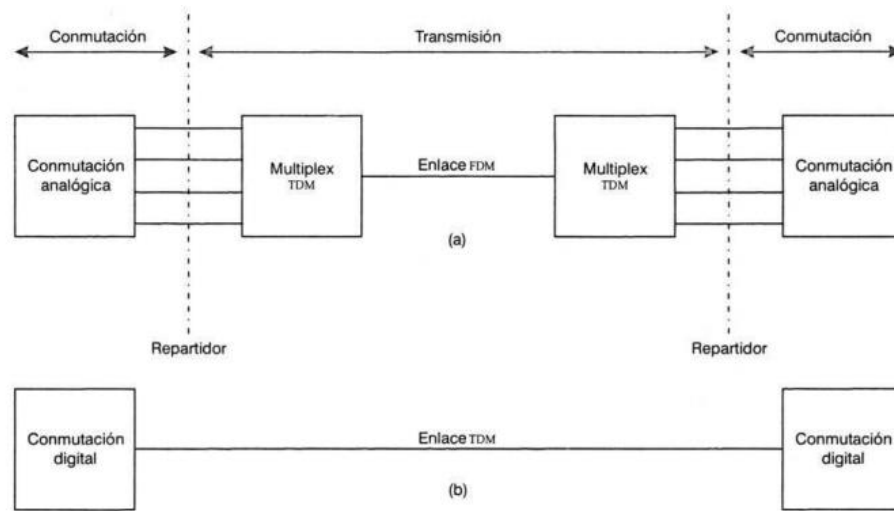


Figura 5 (a) Transmisión y conmutación no integradas; (b) Integración de la transmisión y conmutación

Fuente: (Viveros, 2015)

Funcionamiento con bajas relaciones señal-ruido: "...la facilidad de los sistemas digitales para operar a bajas relaciones señal-ruido es más un requerimiento que una ventaja." (Viveros, 2015, pág. 19). Un ejemplo a tomar, son las señales vocales digitalizadas, las cuales requieren un ancho de banda ocho veces mayor al que utiliza una señal analógica. En caso de utilizar cable, para la transmisión de datos, en él se atenúa la señal, por lo tanto, la operatividad del sistema digital ha de ser con bajas relaciones señal-ruido.

Facilidad de señalización: Una forma de incorporar la información de control en un enlace digital es asignándole un canal común donde se multiplexan las señalizaciones de los diversos canales, otra forma consiste en asignar bits de control junto con el mensaje en cada canal individual. Pero independientemente de los métodos usados, la información de control una vez intercalada se confunde como unos bits más en el flujo total de datos, quedando obviamente a cargo del receptor la tarea de reconocerlos y decodificarlos.

Facilidad criptológica: La seguridad es uno de los requerimientos más comunes en la actualidad; por ello la criptología es "el proceso por el que se modifica un mensaje de forma tal que sólo el destinatario autorizado puede entenderlo." (Viveros, 2015, pág. 20) esta condición de seguridad es implementada a fin de evitar que la información transmitida pueda ser accedida por cualquier usuario del sistema de comunicación digital.

Facilidad de monitoreo: En general la calidad de los sistemas analógicos no puede evaluarse con exactitud mientras el sistema está en servicio. Sin embargo, es posible monitorearlos, por ejemplo, en las señales multiplex FOM suelen insertarse diversos pilotos que permiten medir la continuidad de los canales y la normalidad de los niveles. La medición del ruido y la distorsión puede realizarse evaluándolos en un intervalo vacante dentro de la banda o bien en los bordes de la misma. En ningún caso, sin embargo, tenemos una medición directa de la calidad del canal en servicio.

2.2.3. MULTIPLEXACIÓN POR DIVISIÓN ORTOGONAL EN FRECUENCIA (OFDM).

2.2.3.1. SISTEMA GENERAL DE OFDM.

La Modulación por División Ortogonal de Frecuencias (OFDM) es una de las técnicas más utilizadas en la actualidad porque su esquema de modulación y demodulación es multiportadora, lo cual la hace muy eficaz para utilizarla en aplicaciones que requieren de banda ancha. En un sistema de comunicación mono-portador o Single Carrier (SC), cada uno de los símbolos es transmitido en serie, en otras palabras, uno a la vez, ocupando prácticamente el total del ancho de banda disponible; mientras que en una modulación multiportadora los símbolos son enviados en paralelo a través de subportadoras adyacentes, esto significa que se ahorra espacio en el canal de transmisión, permitiendo una mayor cantidad de información transmitida. (López Cortez, 2015).

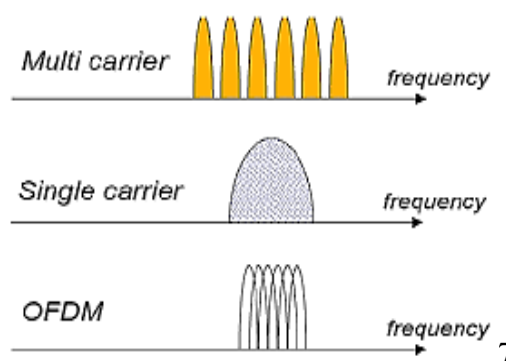


Figura 6 Comparación entre Single Carrier, Multicarrier y OFDM
Fuente: (López Cortez, 2015)

Lo que caracteriza a esta técnica, es el uso de una transformada rápida de Fourier también conocida como FFT (Fast Fourier Transform) con la cual se puede transmitir en las subportadoras paralelas de manera eficaz y eliminando o reduciendo la interferencia entre ellas. (Prieto Del Amo, 2015). Por este motivo, los datos del número de subportadoras están vinculados al número de muestras que utiliza la FFT.

De manera que, la transmisión directa de una trama digital, donde se requiere de una gran tasa de transferencia de NFFT en líneas paralelas más lentas, y en subportadoras ortogonales y contiguas, las cuales transportan

los símbolos independientemente, resultando como producto de un tipo de modulación digital tipo QPSK o 16-QAM, eso depende del sistema. (Masero, 2015).

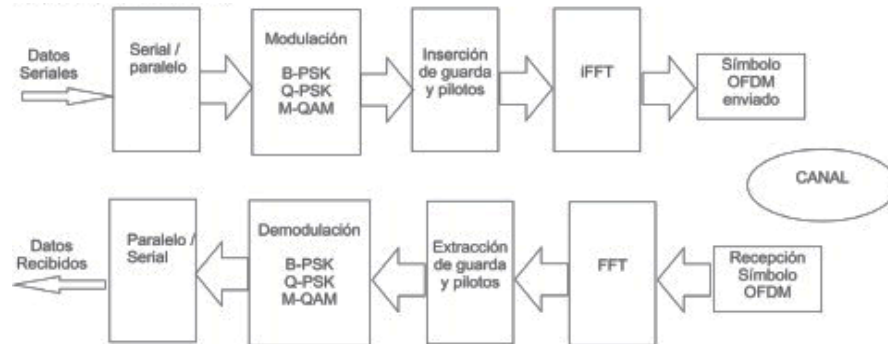


Figura 7 Modelo de un sistema OFDM simple

Fuente: (López Cortez, 2015)

En la ilustración anterior se puede apreciar el proceso de completo a través de un sistema de OFDM simple, donde la señal ingresa al transmisor para ser modulada, donde es dividido en N flujos paralelos y llevados en simultáneo a la FFT para añadir el prefijo cíclico de longitud en los últimos datos de la señal, de ahí pasa a una señal paralela que es llevada en el canal de transmisión seleccionado, hasta llegar al receptor OFDM donde se realiza el proceso inverso, hasta obtener la señal original de la información. Cuando la señal ha pasado por el receptor, esta debería ser semejante a la señal original, dependiendo de las características del canal donde se haya transmitido.

2.2.3.2. USO DE LA FFT EN OFDM.

El problema con el esquema OFDM es que con el fin de obtener una alta resistencia cuando el desvanecimiento se produce en un canal, el tamaño del bloque, N, debe estar en el orden de 100, el cual puede hacer que se requiera una mayor cantidad de módems de subcanal” (López Cortez, 2015, pág. 21). Dicho problema se puede dar una solución mediante la transformada discreta de Fourier, con la cual se logra la simplificación en el hardware de un módem OFDM. Esto ocurre porque la transformada de Fourier descompone una forma de onda en sinusoides de frecuencias diferentes, las cuales al ser sumadas resultan en la forma de onda original. (Prieto Del Amo, 2015). La transformada de Fourier de $f(x)$ está definida por:

$$F(w) = \int_{-\infty}^{\infty} f(x) \exp^{-jx} dx$$

Y su inversa está dada por:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w) \exp^{jwx} dw$$

Pero, la era digital obliga a realizar cambios a la forma tradicional de una FFT para ajustarse a los valores discretos existentes en los actuales sistemas digitales.

Dicha serie modificada es denominada como Transformada Discreta de Fourier (DFT). La cual se obtiene de la siguiente fórmula:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp \frac{-j2\pi kn}{N}$$

Y su inversa está denotada como:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp \frac{j2\pi kn}{N}$$

En un sistema OFDM se utiliza la transformada rápida de Fourier (FFT), la cual es un algoritmo de la FDT que disminuye la cantidad de procesos computacionales en el sistema.

2.2.4. TRANSMISOR Y RECEPTOR OFDM.

2.2.4.1. TRANSMISOR OFDM.

Este dispositivo de transmisión del sistema OFDM convierte un grupo de bits en un conjunto de números complejos que corresponden a valores dentro de las constelaciones que hay disponibles para una modulación I/Q. Además, que posee etapas que son destinadas a la corrección de los diferentes errores de transmisión y la suma del prefijo cíclico. “En general, cualquier procesamiento adicional que deba realizarse con los datos se realiza antes de la IFFT. Esto hace que la modulación OFDM sea apropiada, a la cual llegan solo los símbolos I/Q que están ordenados en la trama que es la que se envía”. (Uruchima Portilla, 2016, pág. 22).

Todo lo anterior se refiere a las “etapas de la adición de los códigos convolucionales o la redundancia cíclica. Se puede incluir la adición de FEC (Forward Error Correction) en virtud del cual se puede llegar a denominar al sistema como COFDM (Coded OFDM)” (Masero, 2015, pág. 22). En esta etapa se cuenta con un número finito de esquemas, de los cuales se escogen algunos para que operen entre los de mayor orden según el estándar a utilizar. En sistemas cableados como el xDSL por lo general se usan aquellos de mayor orden como 256-QAM; en cambio los sistemas inalámbricos utilizan los de menor orden como BPSK o QPSK.

En un transmisor OFDM se utilizan diversos códigos como el Gray y el Entreladado, dependiendo de la tecnología a emplearse; incluso la trama OFDM suele aplicar la inserción de pilotos, en aquellas frecuencias fijas que no se usan para la transmisión de datos, enviando en su lugar secuencias de

bits conocidos con modulaciones en el orden bajo como BPSK o QPSK. Dichos pilotos se utilizan para sincronizar el transmisor y el receptor, así como la estimación del canal a usarse.

2.2.4.2. RECEPTOR OFDM.

Este dispositivo receptor es el que recibe la señal compleja a través del canal I y Q para la realización del trabajo de demodulación, el cual lo efectúa de manera inversa al proceso que realiza el transmisor. A esta señal digitalizada, se le extrae el tiempo de guarda, esto significa para el sistema, la reducción del número de muestras de NFFT, sobre las cuales se aplica la FFT, convirtiendo las muestras en tiempo, en muestras en frecuencia. (Arraño & Azurdia, 2015).

El receptor también tiene la responsabilidad de la sincronización en tiempo y frecuencia o ecualización, y de ser necesario etapas de realimentación para la estimación del canal, para la cual se debe valerse de las secuencias de datos conocido, así como los pilotos insertados en los datos que se envían en las tramas. Finalmente, se realizan las etapas de demodulación, desentrelazado y de detección de errores.

2.2.5. TIEMPO DE GUARDA Y PREFIJO CÍCLICO.

En OFDM Se inserta un tiempo de guarda entre los símbolos OFDM para evitar la interferencia entre los símbolos (ISI) y la interferencia entre los canales (ICI) con multitrayectoria. El prefijo cíclico tiene una longitud de N_g muestras, igual p según la teoría es mayor que el rango de retardo máximo del canal. El prefijo cíclico provoca una convolución lineal de la respuesta de impulso del canal y la señal se convierte en un bucle circular.

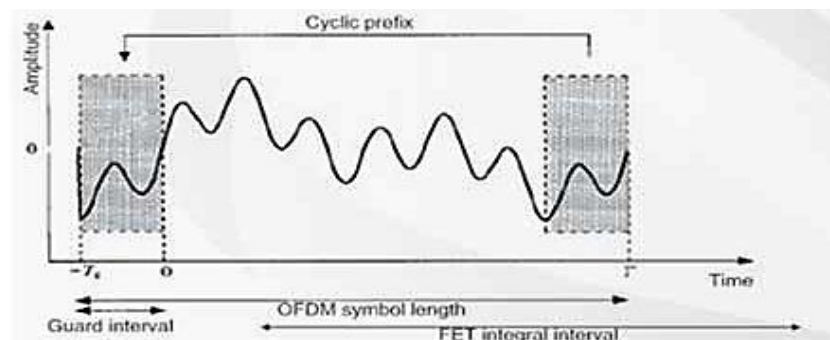


Figura 8 Intervalo de guarda y prefijo cíclico
Fuente: (Arraño & Azurdia, 2015)

Básicamente, se lo utiliza como una banda de guarda para cada símbolo. Consiste en realizar una “réplica” de la última parte de la señal del símbolo OFDM y colocarlo al inicio, como se muestra en la ilustración. Para OFDM, este es básicamente un estándar que usa el prefijo cíclico durante este intervalo. De esta manera, el símbolo OFDM se completa al agregar una muestra que emite un IFFT antes de la NFFT. Luego debe distinguir entre el tiempo útil T_u y el tiempo de guarda T_g dentro del símbolo T_{sym} . Como

primer resultado, esto reduce la eficiencia de transmisión, pero el beneficio es mayor al mitigar los efectos de la demora en los canales de múltiples rutas. De acuerdo con lo anterior, el tiempo del símbolo OFDM es,

$$T_{sym} = T_u + T_g$$

Y para el tiempo de guarda es,

$$T_g = G \times T_s$$

Además de dejar el intervalo de guarda a cero o usar un prefijo cíclico, hay una mayor probabilidad de trabajar con intervalos de guarda con un cambio en la longitud de la longitud. En la otra mitad de la semana, puede mencionar el uso de una señal de cero para usar una ocupación periódica del 50% del tiempo de guardia. Este tipo de implementación proporciona beneficios de temporización del sistema, pero no proporciona mucha protección para canales con múltiples rutas.

2.2.6. MODELO DE SISTEMA DE TIEMPO DISCRETO OFDM.

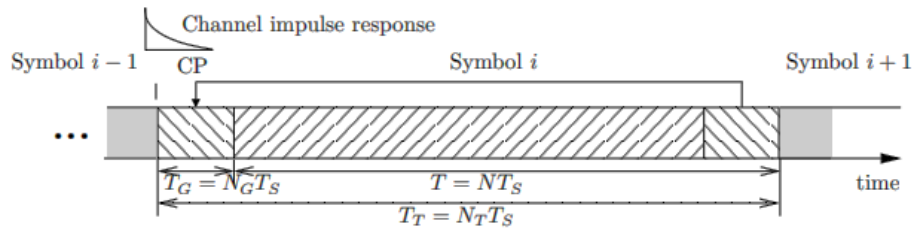


Figura 9 Estructura de un símbolo OFDM
Fuente: (Masero, 2015)

Como OFDM es un modelo de comunicación basado en bloques, un flujo de datos en serie se convierte en bloques paralelos de tamaño N muestras e IDFT se aplica para obtener símbolos OFDM de dominio de tiempo. Símbolos de datos complejos $C_i(n)$, para $n=0, \dots, N-1$, dentro del i -ésimo símbolo OFDM se toman de una Modulación por desplazamiento de Fase (PSK) o Constelación de Modulación de Amplitud en Cuadratura (QAM). Entonces, la representación del dominio del tiempo del i -ésimo símbolo OFDM después de la inserción de IDFT y CP (prefijo cíclico), está dado por:

$$c_i(k) = \begin{cases} \sum_{n=0}^{N-1} C_i(n) e^{j2\pi kn/N}, & -N_g \leq k \leq N-1, \\ 0, & \text{else} \end{cases}$$

Donde N_g es la longitud del CP que es un parámetro de diseño importante del sistema OFDM que define la duración máxima aceptable de la respuesta al impulso del canal. la señal transmitida puede ser obtenida por concatenación de los símbolos OFDM en el dominio del tiempo de esta forma:

$$c(k) = \sum_1 c_i(k - iN_T) \quad (2.2)$$

En los sistemas de comunicación inalámbrica, las señales transmitidas son típicamente reflejadas, difractadas y dispersa, llegando al receptor a lo largo de múltiples caminos con diferentes retrasos, amplitudes y fases. Esto conduce a una superposición de diferentes copias de la misma señal en el lado del receptor que difieren en su amplitud, tiempo de llegada y fase. Un modelo común para describir el canal inalámbrico hace uso de la respuesta al impulso del canal escrito como $h(i) = \alpha(i)e^{j\theta(i)}$, para $i=0, \dots, L-1$, donde L representa el número total de trayectos de señal recibidos, mientras que α_i y θ_i son atenuación y desplazamiento de fase del i -ésimo camino, respectivamente. Las diferencias en el tiempo de llegada se eliminan por el prefijo cíclico que se describe en siguientes secciones.

Además de los efectos multitrayecto, se introduce ruido aditivo en la señal transmitida. Las principales fuentes de ruido aditivo son el ruido de fondo térmico, el ruido eléctrico en los amplificadores del receptor. El ruido disminuye la relación señal/ruido (SNR) de la señal recibida, lo que resulta en una disminución del rendimiento. El ruido efectivo total en el receptor de un sistema OFDM se puede modelar como AWGN con una densidad espectral uniforme y una probabilidad gaussiana de distribución zero-mean. Las muestras de ruido en el dominio del tiempo están representadas por $w(k) \sim \text{SCN}(0, \sigma_w^2)$, donde σ_w^2 denota la varianza del ruido y cero la media de la distribución compleja circular simétrica. Por lo tanto, el modelo de señales OFDM recibido para un tiempo discreto puede ser escrito como:

$$y(k) = \sum_{i=0}^{L-1} h(i)c(k - i) + w(k) \quad (2.3)$$

Propagación multitrayecto y el ruido aditivo afectan la señal de manera significativa, corrompiendo la señal y a menudo dando lugar a limitante en el diseño del sistema.

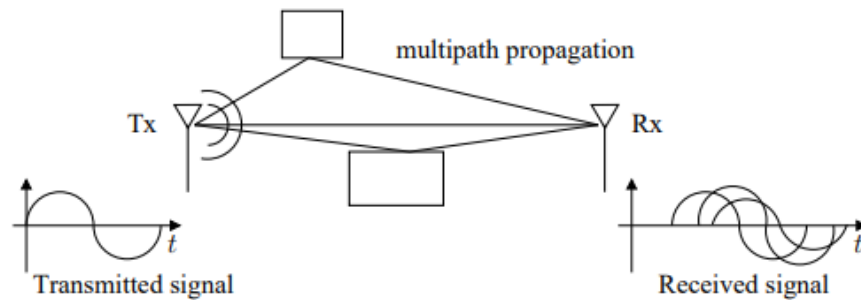


Figura 10 El principio básico de la propagación multitrayecto.

Fuente: (Masero, 2015)

2.2.7. DEFICIENCIAS DEL SISTEMA OFDM.

Dado que los errores de temporización y frecuencia en los sistemas multiportadora destruyen la ortogonalidad entre las subportadoras, resulta en grandes degradaciones de rendimiento, la sincronización de tiempo y frecuencia juega un papel relevante en el diseño de un sistema de comunicación digital. Esencialmente, esta función tiene como objetivo recuperar algunas referencias parámetros de la señal recibida que son necesarios para la detección confiable de datos. En un sistema OFDM la sincronización del reloj de muestreo en sistemas prácticos, la frecuencia del reloj de muestreo en los receptores ligeramente diferente de la frecuencia correspondiente en el transmisor. Esto produce interferencia entre portadoras (ICI) en la salida del DFT del receptor con una degradación correspondiente del rendimiento de sistema. El propósito de una sincronización de reloj de muestreo es limitar este deterioro a un nivel tolerable.

Otro objetivo es la sincronización del tiempo, que permitirá identificar el punto de partida de cada operación Símbolo OFDM para encontrar la posición correcta de la ventana DFT. En transmisiones en modo ráfaga sincronización de tiempo también se utiliza para localizar el inicio de la trama (sincronización de cuadro) que es una colección de símbolos OFDM.

Un error de frecuencia entre los osciladores locales en el transmisor y el receptor da como resultado la pérdida de ortogonalidad de las subportadoras con las consiguientes limitaciones del sistema actuación. La sincronización de frecuencia tiene como objetivo restaurar la ortogonalidad mediante la compensación de cualquier desplazamiento de frecuencia causado por inexactitudes del oscilador.

En la interfaz analógica, la forma de onda entrante $T_{RF}(t)$ se filtra y se convierte a la banda base utilizando dos sinusoides en cuadratura generadas por un local, oscilador (LO). La señal de banda base se pasa luego al convertidor de analógico a digital (ADC), donde este es muestreado con frecuencia $F_s = 1/T_s$.

Debido a los cambios Doppler y / o las inestabilidades del oscilador, la frecuencia f_{LO} de la DFT no es exactamente igual a la frecuencia portadora recibida f_c . La diferencia $f_d = f_c - f_{LO}$ es denominada desplazamiento de frecuencia de portadora (CFO), o desplazamiento de frecuencia más corto, que causa un desplazamiento de fase de $2\pi k f_d$. Además, desde las escalas de tiempo en el transmisor y el receptor no están perfectamente alineados, en la puesta en marcha el receptor no sabe dónde comienzan los símbolos OFDM y, en consecuencia, la ventana DFT se colocará en una posición incorrecta. Como se mostrará más adelante, ya que los errores de temporización pequeños (fraccionarios) no producen ninguna degradación del rendimiento del sistema, es suficiente para estimar el comienzo de cada símbolo OFDM recibido dentro de un período de muestreo. Dado que Δk

denota el número de muestras por las cuales la escala de tiempo de recepción se desplaza de su configuración ideal. Las muestras del ADC están expresadas por:

$$r(k) = e^{j2\pi k/N} y(k - \Delta k) + w(k). \quad (2.6)$$

Sustituyendo (2.2) y (2.3) en (2.6), las muestras se dan como:

$$r(k) = e^{j2\pi k/N} \sum_i \sum_{l=0}^{L-1} h(l) c_i(k - l - \Delta k - iN_T) + w(k)$$

Las unidades de sincronización de frecuencia y tiempo emplean las muestras recibidas $r(k)$ para calcular las estimaciones de ε y Δk , indicadas como ε y Δk . El primero se usa para contrarrestar $r(k)$ a una velocidad angular $2\pi\varepsilon k / N$ (corrección de frecuencia) usando un oscilador controlado numéricamente (NCO), mientras la estimación de tiempo se aprovecha para lograr la posición correcta de la señal recibida dentro de la ventana DFT (corrección de tiempo). Específicamente, las muestras $r(k)$ con índices $iN_T + \Delta k \leq k \leq iN_T + \Delta k + N - 1$ son alimentados al dispositivo DFT y la salida correspondiente se utiliza para detectar los símbolos de datos transmitidos por el i -ésimo bloque OFDM.

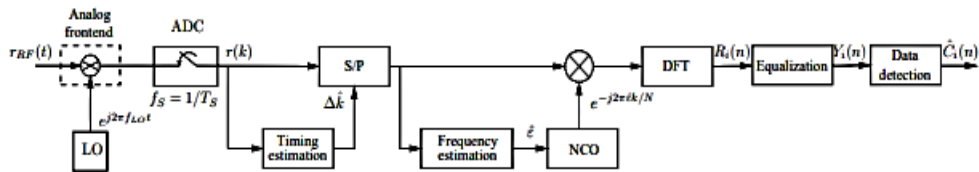


Figura 11 Diagrama de bloques de un Receptor OFDM.

Fuente: (Masero, 2015)

2.2.8. VENTAJAS DE OFDM.

Una de sus principales ventajas es la resistencia al desvanecimiento por un canal con selectivo desvanecimiento de frecuencia debido a un tiempo de guarda apropiado. Ocurre en un entorno donde no hay una línea de visión o un ojo problemático. Esto no es necesario para que el receptor tenga una antena direccional fuerte como de costumbre.

La técnica OFDM posibilita el diseño de un sistema de comunicaciones con una amplia tasa de transmisión de datos evitando la interferencia entre símbolos. También la reducción de los procesos computacionales por el uso de FFT y IFFT ahorrando recursos de hardware al sistema. Además, permite un eficiente uso del espectro, debido a la utilización de subcanales dentro del ancho de banda, favoreciendo la transmisión frente a los desvanecimientos selectivos en frecuencia. Incluso la utilización del prefijo cíclico es una mejora en la transmisión de datos.

Desde el punto de vista técnico de implementación y uso de recursos se cuenta con la mejor eficiencia espectral y se conseguiría con las técnicas SC

y la implementación en la FFT, la cual reduce la complejidad de transmisor y receptor. La versatilidad de este sistema permite la implementación de muchas tecnologías complementarias que se proyectan a mejorar su funcionamiento mundial y aprovechar aún mejor el espectro, esto se consigue mediante técnicas combinadas.

3. MARCO METODOLÓGICO

3.1. TIPO DE INVESTIGACIÓN.

El presente proyecto, se realiza basándose en una investigación de tipo exploratoria tomando en consideración a (Facal, 2015), el cual manifiesta que la modalidad exploratoria “busca la visión general o amplia de una realidad determinada, misma que se singulariza por el carácter preparatorio que posee” (Facal, 2015, pág. 23). Siguiendo estas palabras, se realiza una investigación exploratoria porque las variables de estudio no se estudian de manera práctica en la Universidad Politécnica Salesiana Sede Guayaquil; por este motivo, es necesario realizar un análisis de algunos aspectos relacionados con el uso de señales en sistemas OFDM.

También, es experimental porque “se influye activamente en algo para observar sus consecuencias” (Marcano, 2018, pág. 4), es decir, se realiza una práctica experimental en uno de los laboratorios de la carrera Ingeniería Electrónica de la Universidad.

Durante la ejecución de las siete pruebas, configuran los equipos, se aplican modulaciones de frecuencias, sincronización de señal y análisis de resultados. Por lo tanto, es descriptiva porque se van describiendo los resultados de cada una de las prácticas. (Universia Costa Rica, 2017).

3.2. DISEÑO DE INVESTIGACIÓN.

En diseño de la investigación aplicado es de un estudio tipo experimental, porque de acuerdo a (Arias, 2013), “este permite la manipulación de un objeto o a un grupo de personas para someterlos a diferentes condiciones y estímulos para realizar la observación de los resultados”. (Arias, 2013, pág. 41).

Este estudio se basa en información extraída de diferentes artículos científicos con la misma temática o similar a ella; además, el estudio se basa en la programación de bloques OFDM, mismos que podrían variar posteriormente luego de su configuración en el sistema y añadir mayores funciones al proyecto, la cual se irá verificando en las pruebas realizadas.

3.3. ENFOQUE DE LA INVESTIGACIÓN.

En esta investigación de campo, se aplica un enfoque mixto, es decir, cuali-cuantitativo porque se consideran a las variables en dos partes fundamentales; como la revisión teórica, la cual se realiza por medio de la

investigación bibliográfica de diversas fuentes como: libros, tesis relacionadas, artículos científicos y demás publicaciones, donde se realiza el análisis cualitativo de la información obtenida.

Por otra parte, se realiza un estudio experimental, en otras palabras se realiza una investigación de campo, donde se obtienen datos cuantitativos, debido al análisis de las diferencias entre la modulación y demodulación de señales en un sistema OFDM, lo cual permite que el estudio de estas se realice con un máximo de aprendizaje práctico, aparte del teórico. (Cadena Iñiguez, 2017).

3.4. METODOLOGÍA DE LA INVESTIGACIÓN.

En cuanto al método de investigación, se trabaja con un método inductivo, el cual, de acuerdo con (Rodríguez, 2013), busca la inducción del pensamiento a través de la generación de ideas particulares que en conjunto forman una generalidad del pensamiento, otorgando la posibilidad de obtener conclusiones relacionadas al estudio.

En este caso, a partir de configuraciones y pruebas realizadas y el análisis de los resultados obtenidos se logra establecer conclusiones generales para el diseño de algoritmos para la sincronización y codificación de canales de sistemas OFDM utilizando radios definidos por software.

3.5. INSTRUMENTOS DE INVESTIGACIÓN.

Para el presente trabajo investigativo se utilizaron materiales e instrumentos como los mencionados a continuación:

3.5.1. GNU RADIO.

3.5.1.1. ESTRUCTURA DE BLOQUES EN GNU RADIO.

Las aplicaciones de GRC (GNU Radio Companion) se escriben primordialmente utilizando lenguaje de programación Python, mientras que para el procesamiento de señales esta implementado en C++. Como mencionamos con anterioridad, modificar parámetros de operación en el software es más redituable que realizar actualizaciones de hardware.

La creación de bloques de procesamiento en GRC desarrolla un papel importante en el campo de los SDR ya que permiten implementar una gran variedad de algoritmos para modificar el funcionamiento al simular o emular un estándar de telecomunicaciones.

Las aplicaciones en GNU Radio se escriben con una combinación de los lenguajes Python y C++, es utilizado para la construcción de bloques jerárquicos y la conexión entre bloques de procesamiento, mientras que C++ es utilizado para la construcción de bloques de procesamiento. SWIG (SimplifiedWrapper and InterfaceGenerator) es el intérprete entre Python y

C++. GNU Radio funciona sobre las distintas plataformas como Linux, Mac y Windows, en próximos capítulos indicamos los pasos para la correcta instalación de esta herramienta, en nuestro proyecto la instalamos en Linux.

En la siguiente figura podemos observar las capas que componen el sistema SDR.

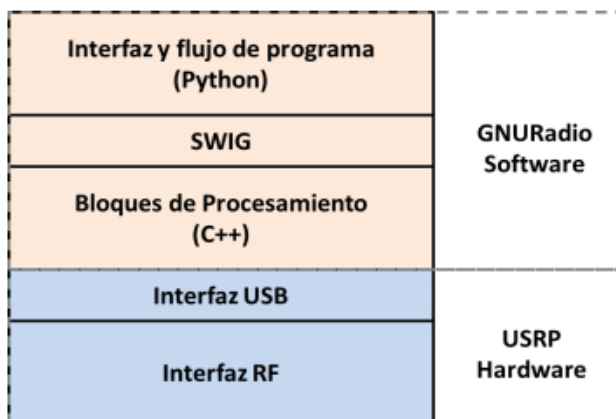


Figura 12 Componentes del esquema SDR

Fuente: (Cadena Iñiguez, 2017)

Se utiliza también GNU Radio-companion (GRC) que es la interfaz gráfica en la cual el desarrollador puede adquirir beneficio para crear una aplicación GNU Radio completa sin la necesidad de escribir código, se trata de una herramienta de graficación opcional para la simplificación del nivel de complejidad y de los conocimientos de lenguajes de programación.

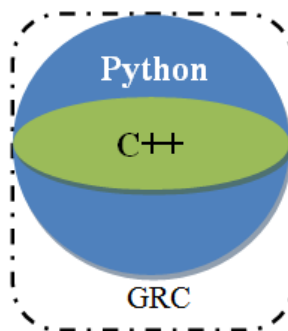


Figura 13 Niveles de abstracción GNU Radio

Fuente: (Cadena Iñiguez, 2017)

A continuación, se presenta la manera en que se agrupan las librerías y los archivos que conforman el GNU Radio.

Dichas librerías y los archivos son puestos en grupos en módulos dependiendo de la debida función que estos desempeñen, los módulos principales son:

Módulos GNU Radio	
Gr	Es el módulo principal del GNU radio, este se va a necesitar prácticamente para todos los casos ya que contiene los bloques básicos, así como la gran parte de las fuentes, la gran parte de los distintos sumideros, así como adición, sustracción, entre otros.
Digital	Este módulo es el que contiene las librerías y los archivos que asumen llevar a cabo las modulaciones y las demodulaciones digitales.
Audio	Este módulo es el que proporciona el control sobre la tarjeta de sonido, permite el envío o recepción de las señales de audio usando la tarjeta de sonido.
Blocks	Este módulo es el que contiene los bloques de procesamiento utilizados en los Flow graphs.
blks2	Este módulo contiene bloques adicionales escritos en Python.
Trellis	Este módulo le provee los archivos que son necesarios para la realización de codificaciones convolucionales.
Analog	En este módulo es donde se ubican los distintos archivos relativos de las modulaciones analógicas.
Wavelet	Este módulo proporciona bloques de procesamiento para las transformadas wavelets.
Fft	Este módulo proporciona bloques de procesamiento para las Fast Fourier Transform(FFT)
Window	Contiene las rutinas de diseño para las ventanas.
Optfir	Este módulo contiene las rutinas de diseño óptimo de los filtros de respuesta al impulso finito (FIR)
Filter	Este módulo es el que proporciona los bloques de procesamiento para las operaciones de filtrado.
Qtgui	ES el que proporciona sumideros gráficos basados en QT.
Wxgui	Módulo que proporciona una GUI basada en Wx.
Grc	Módulo necesario para poder utilizar la interfaz gráfica gnuradio-companion.
video_sdl	Este módulo proporciona el control para el envío y recepción de señales de video.
Vocoder	Este módulo incluye los variados bloques para el procesamiento los cuales implementan vocoders.
Uhd	Este es el módulo que sirve para la interfaz de la librería UHD para la transmisión y recepción de datos en los USRP.
Howtowrite a block	Este es el módulo que contiene la información para la creación de los nuevos módulos y así incluirlos al proyecto del gnuradio.

Tabla No. 1 Módulos GNU Radio

Los módulos del GNU Radio son estructurados por carpetas, dichas carpetas son encargadas de agrupar las mencionadas librerías y los archivos, comúnmente un módulo de GNU Radio tiene la siguiente estructura:

Estructura de un módulo GNU radio	
Carpetas	Descripción.
Apps	Esta carpeta contiene ejemplos y aplicaciones de prueba del módulo.
Cmake	Esta carpeta contiene archivos de configuración necesarios para la correcta instalación del módulo.
GRC	Esta carpeta contiene los diferentes archivos “.xml” de los bloques para poder usarlos en la aplicación GNU Radio-companion.
Include	Esta carpeta contiene los archivos fuente de las librerías “.h” de los bloques de procesado.
Lib	Esta carpeta contiene los archivos fuente “.cc” de los bloques de procesado.
Python	Esta carpeta contiene los diferentes scripts de Python.
Swig	Contiene los archivos swig “.i” con la configuración del intérprete de C++ y Python.

Tabla No. 2 Estructura de un módulo GNU Radio

3.5.2. GNU RADIO-COMPANION.

GNU Radio Companion es un conjunto de herramientas de desarrollo de software libre y de código abierto que provee de algoritmos en bloques de procesamiento para implementar sistemas de radiocomunicación. Se puede utilizar con hardware externo de bajo costo o simplemente como entorno de simulación.

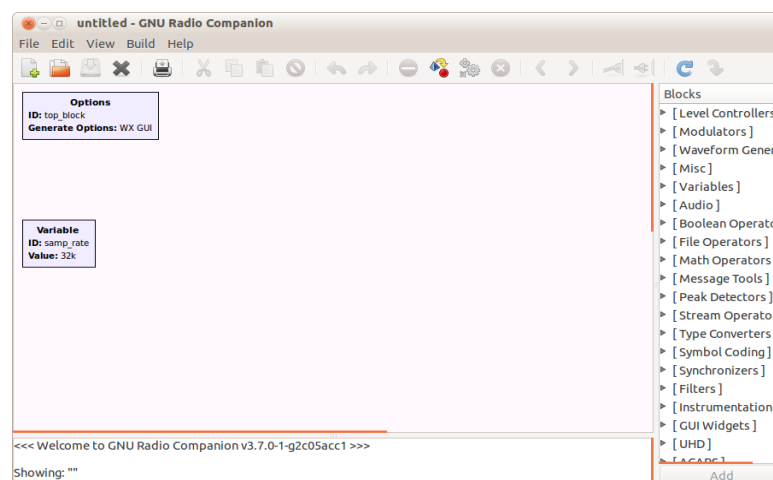


Figura 14 Interfaz de trabajo GNU Radio-companion

Fuente: (Escobar, 2016)

Es ampliamente utilizado por aficionados, sectores académicos y comerciales para apoyar tanto a los sistemas de radio en el mundo real y también en la investigación de comunicaciones inalámbricas.

Para añadir un bloque en el esquemático solamente daremos doble clic sobre el que deseamos añadir y para interconectar bloques, bastará simplemente con seleccionar los bloques a unir en secuencia del orden que vaya.

3.5.3. PYTHON.

Python es uno de los lenguajes de programación de nivel alto y de código abierto creado por Guido van Rossum y publicado en 1991. Es un lenguaje interpretado, es decir, el código se ejecuta por un intérprete. Debido a esto los archivos de código fuente en Python suelen ser llamados comúnmente scripts, si bien estos archivos reciben oficialmente el nombre de módulos. Los módulos pueden ir agrupados en paquetes.

Python está diseñado para que sea legible y limpio. Adicionalmente no permite programar en varios paradigmas de programación: programación orientada a objetos, funcional, imperativa. Los principios diferenciadores de este lenguaje son:

- Sintaxis clara y legible.
- Fuerte capacidad de introspección en el funcionamiento del mismo.
- Orientación a objetos intuitiva.
- Expresión natural de código procedimental.
- Modularidad total, soportando jerarquía de paquetes.
- Tratamiento de errores mediante excepciones.
- Tipos de datos dinámicos de muy alto nivel.
- Librerías estándar completas y módulos de terceros para prácticamente cualquier tarea.
- Fácilmente extensible mediante código en C/C++ u otros lenguajes.
- Empotrable en aplicaciones como una interfaz de scripting.

3.5.4. RADIO DEFINIDO POR SOFTWARE.

SDR (Radio definido por software) consiste en un sistema de comunicaciones por radiofrecuencia, en el cual la parte de hardware como multiplexores, filtros digitales, amplificadores, moduladores de señal, codificadores etc. es implementada a través de programación en software, mediante dispositivos electrónicos, generalmente basados en sistemas embebidos.

De acuerdo con la IEEE (Institute of Electrical and Electronics Engineers) radio definido por software es definido como “Radio en el que todas o algunas de sus funciones son definidos por software”. SDR es un conjunto de tecnología como tecnología software, tecnología de hardware de procesamiento de señal, tecnología RF (Radiofrecuencia), etc. Cuando usamos

este tipo de tecnologías, implementamos un receptor Zero-IF o low-IF1 configurable de tal manera que puede ser utilizado para diseñar distintos componentes que hacen aplicable el concepto de SDR a los sistemas completos tales como transmisores, receptores, transceptores, osciloscopios, analizadores de espectros o analizadores vectoriales de redes, siendo sus parámetros configurables dinámicamente.

3.5.4.1. ESTRUCTURA.

Con el crecimiento de SDR, se pudo lograr facilitar el diseño de la etapa frontal de radiofrecuencia del radio para obtener una señal que pueda ser fácilmente tratada por un computador, para lograr esto se implementaron conversores de analógico a digital y viceversa en procesadores programables de alto desempeño como los FPGA con el objetivo de reducir el tamaño y costo de los circuitos, sin embargo, el esquema básico de funcionamiento sigue siendo el mismo.

- Sección de RF, implementada en Hardware
- Sección de IF, implementada en Hardware
- Sección banda base, implementada en Software.

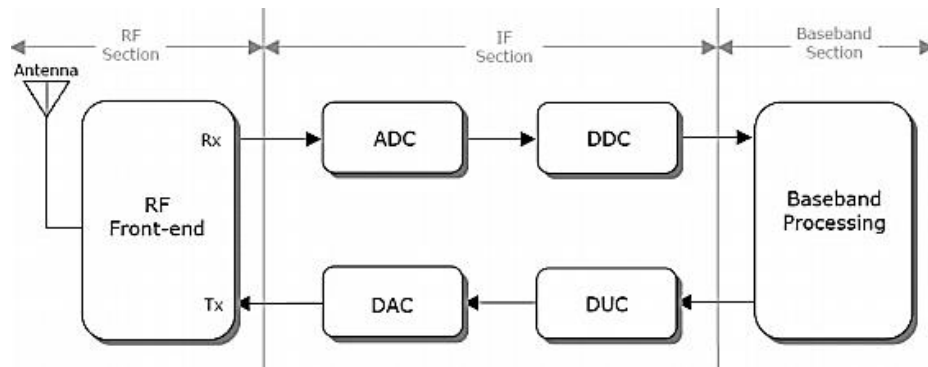


Figura 15 Diagrama de bloques de SDR
Fuente: (Cadena Iñiguez, 2017)

La sección de RF, también denominada RF Front-End es la encargada de transmitir/recibir las señales de radiofrecuencia para adaptarlos y convertirlos de recepción a frecuencia intermedia, o amplifique y module la señal de FI en el caso de transmisión. La frecuencia intermedia puede ser cero y genera el concepto de cero-IF. Esto se debe a los avances en componentes de hardware. De manera similar, la parte IF transporta la señal IF a la banda base, la digitaliza en el receptor o pasa la señal de banda base a la IF y realiza una conversión digital a analógica de la señal en el caso de la transmisión.

El módulo DDC/DUC se inserta para disminuir o aumentar la frecuencia de muestreo en la dirección de recepción/transmisión, respectivamente, lo que resulta en una menor frecuencia de muestreo a través de la interfaz entre IF y banda base. La sección de banda base es responsable de todo el procesamiento de la señal en banda base, como la

modulación/demodulación de las señales llevadas a cabo en el software, el análisis espectral.

3.5.4.2. UNIVERSAL SOFTWARE RADIO PERIPHERAL (Sistema USRP).

El uso de software o hardware especializado para el procesamiento de señales de radiofrecuencia permite una mayor experimentación en escenarios de pruebas reales.

En la mayoría de los casos se vuelve muy demandante ejecutar algoritmos para el procesamiento de señales utilizando solamente software como Matlab, por ejemplo, por esto es necesario el uso de hardware para la implementación de funciones como FFT o IFFT (Inverse Fast Fourier Transform) en esquemas de OFDM. El uso de una plataforma SDR nos permite validar resultados esperados obtenidos típicamente a través de simulaciones. En la figura podemos observar de manera muy general la implementación de un sistema de telecomunicaciones usando GNU Radio y hardware USRP.

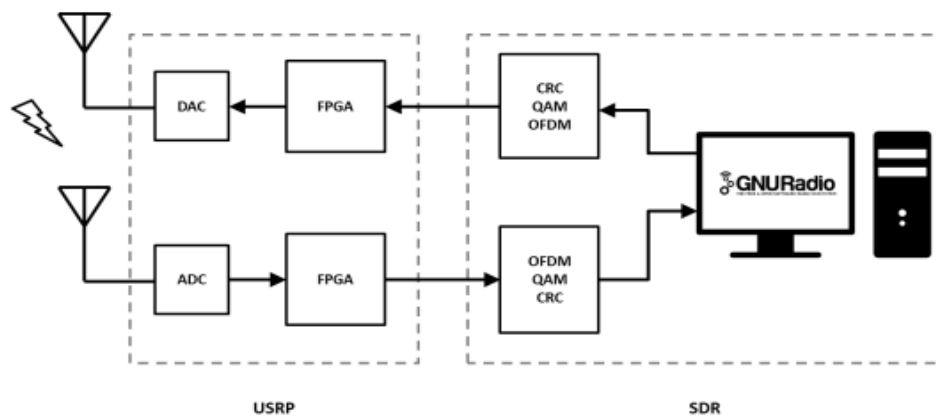


Figura 16 Sistema de Radio Definido por Software
Fuente: (Cadena Iñiguez, 2017)

Las tarjetas USRP son tecnología basada en SDR desarrollada por la compañía Ettus Research y su empresa matriz National Instruments. El proyecto inicialmente fue concebido con el fin de que se convierta en una plataforma de bajo costo para el diseño de aplicaciones de radiofrecuencia en centros de investigación, laboratorios y universidades. Los USRPs se conectan a una computadora de propósito general mediante una interfaz USB o Ethernet por la cual el software puede transmitir y recibir datos. Las tarjetas USRP son utilizadas en conjunto con un controlador UHD (USRP Hardware Driver) de código abierto y el software GNU Radio para crear sistemas de radio basados en SDR.

3.5.4.3. CARACTERÍSTICAS.

El USRP internamente presenta dos niveles de tarjetas sin importar el fabricante. El primero es la tarjeta principal, también denominada madre o motherboard, en donde podemos encontrar el FPGA, los convertidores

ADC's y DAC's, la alimentación y la conexión vía USB. El segundo nivel consiste en tarjetas hijas, hijas o tablas hijas. Estos existen para la transmisión y / o recepción.

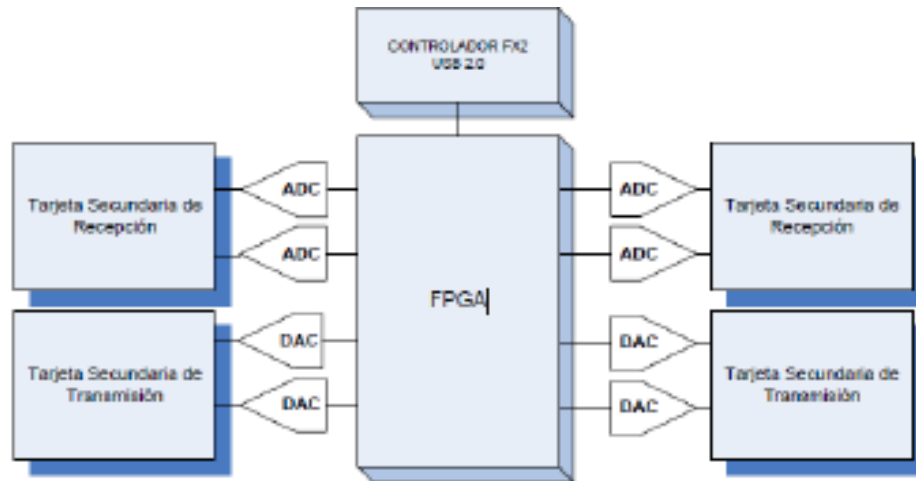


Figura 17 Diagrama de bloques de la estructura interna de un USRP

Fuente: (Cadena Iñiguez, 2017)

De esta manera, el USRP puede operar como múltiples transmisores, receptores o múltiples tarjetas hijas capaces de operar como transceptores (transceptores, TRX), que en este último caso pueden transmitir y recibir simultáneamente y transmitir la banda base o la señal de IF a la RF deseada Banda, o, al contrario.

En la ilustración se muestra un diagrama de bloques del USRP, donde se puede apreciar la disposición de FPGA, convertidores, tarjetas secundarias y la interfaz USB o Gigabit Ethernet. El hardware USRP posee un controlador de dispositivo UHD, proporcionado por Ettus Investigation para su uso con los diferentes tipos de hardware USRP desarrollados hasta hoy: compatible con Linux, MacOS y plataformas Windows. UHD ofrece portabilidad y es amigable con las diferentes plataformas de desarrollo de software:

- GNU Radio. Es un conjunto de herramientas de código abierto, que se puede utilizar para desarrollar SDR. Este marco utiliza una combinación de C++ y Python para optimizar el rendimiento DSP mientras que proporciona un entorno de programación de aplicaciones de fácil uso. GNU Radio es un entorno de programación gráfica.
- National Instruments. En base a su producto, el NI USRP 292X, es funcionalmente equivalente a la Ettus Investigation USRP N210, USRP B205 mini-i.
- USRP N210 y USRP B205 mini-i. dispositivos compatibles con MATLAB y Simulink, herramientas de software que incluyen plug-ins y varios ejemplos de uso con los dispositivos. Muchos usuarios desarrollan con sus propios marcos personalizados.

3.5.4.4. APLICACIONES.

Tiene una extensa lista de aplicaciones reales, partiendo del principio de radiofrecuencia, esto quiere decir, el desarrollo de sistemas de comunicaciones inalámbricas en los cuales se pueda insertar los principios de funcionamiento de este software. A continuación, algunas de las aplicaciones del USRP:

- Sistema de radares.
- Red de telefonía móvil GSM.
- Transmisor y receptor de radio FM.
- Radiodifusión de audio digital.
- Redes de alta velocidad.
- Estándar de televisión digital.
- Analizador de espectros.
- Aplicaciones con GPS.

3.5.5. CARACTERÍSTICAS GENERALES DEL HARDWARE PROPUESTO EN LA SOLUCIÓN.

USRP B205mini-i.

La serie de bus USRP proporciona una plataforma periférica de radio de software única y universal integrada con cobertura de frecuencia continua de 70 MHz a 6 GHz. Diseñado para la experimentación de bajo costo, combina un transceptor de conversión directa totalmente integrado que proporciona hasta 56MHz de ancho de banda en tiempo real, un Spartan6 FPGA abierto y reprogramable, y una conectividad SuperSpeed USB 3.0 de bus rápida y conveniente.



Figura 18 USRP B205mini-i
Fuente: (Ettus, 2017)

- Xilinx Spartan 6 FPGA
- Rango RF de 70 Mhz a 6 GHz
- Interfaz USB 3.0
- 56 MHz de ancho de banda
- Potencia de salida > 10 dBm

El modelo B205mini-i fue elegido para utilizarse en el proyecto ya que tiene las características necesarias para esta tesis. Cuenta con un bus de datos con suficiente ancho de banda (56 MHz), amplio rango de frecuencias (70 Mhz – 6 GHz), teniendo así una plataforma completa para las diferentes aplicaciones que se requieran.

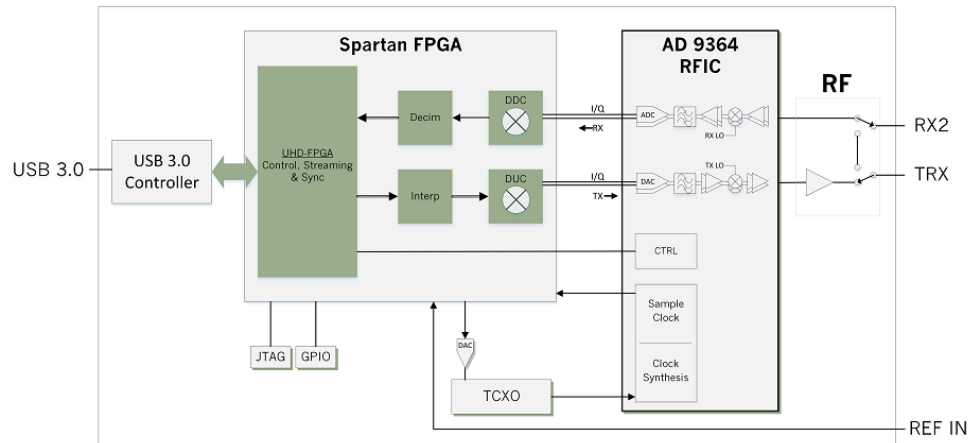


Figura 19 Arquitectura USRP B205 mini-i

Fuente: (Ettus, 2017)

Sintonización: La interfaz de RF tiene cadenas de recepción y transmisión ajustables individualmente. En los mini B200, hay un frontend RF de transmisión y uno de recepción.

Ganancias: Todas las interfaces tienen controles de ganancia analógicos individuales. Las interfaces de recepción tienen 76 dB de ganancia disponible; y las interfaces de transmisión tienen 89.8 dB de ganancia disponible. Las configuraciones de ganancia son específicas de la aplicación, pero se recomienda que los usuarios consideren usar al menos la mitad de la ganancia disponible para obtener un rango dinámico razonable.

Anchos de banda: La interfaz analógica tiene un ancho de banda ajustable de 200 kHz a 56 MHz. La propiedad para controlar el ancho de banda analógico RX es ancho de banda / valor.

Especificaciones RF: Los USRP B205mini se derivan de los dispositivos analógicos AD936x chip transceptor integrado, el rendimiento general de RF del dispositivo se rige en gran medida por el propio chip transceptor.

Rendimiento RF:

Supresión SSB / LO -35/50 dBc

Nivel de ruido 3.5 GHz 1.0 grado RMS

Nivel de ruido 6 GHz 1.5 grados RMS

Potencia de salida > 10dBm

IIP3 (@ typ NF) -20dBm

Figura típica de ruido <8dB

Potencia máxima de entrada: 0 dBm

Impedancia de entrada / salida

Todos los puertos de RF se adaptan a 50 ohmios con una pérdida de retorno de -10dB o mejor en general.

Niveles de potencia de entrada

La potencia máxima de entrada para el B205mini es 0 dBm.

3.5.6. UBUNTU.

Es un sistema operativo de software de código abierto que se puede ejecutar en el escritorio, en la nube, y en múltiples dispositivos y enlaces conectados a internet. Se escogió este sistema operativo por ser operativo desde el escritorio y permitir el desarrollo mediante las tecnologías OFDM. Es una de las distribuciones de Linux que está basada en arquitectura Debian. Esta es la rama del código principal del proyecto Debian.

Su propósito principal es hacer que la distribución Debian sea más sencilla de manejar y más fácil de comprender para el usuario, corrigiendo varios de los errores y facilitando las tareas, como la administración de los programas.

La interfaz predeterminada por Ubuntu es llamada Unity y es la que utiliza en conjunto las aplicaciones de GNOME. Ubuntu divide internamente el software en cuatro, estas secciones se las llama componentes, que muestran diferencias entre las licencias y prioridades de problemas que son informados por los diferentes usuarios. Dichos componentes son main, restricted, universo y multiverse.

De forma predeterminada se instalan los paquetes de estos componentes *main* y *restricted*. Los paquetes del componente llamado *universo* de Ubuntu se basan en paquetes de la rama inestable (Sid) y en el repositorio experimental de Debian.

Los siguientes comandos son necesarios para poder realizar nuestro proyecto:

Comando	Descripción	Ejemplos
Mancomando	Se encarga de mostrar el manual de un comando, que es útil para comprender el uso de argumentos.	Manman

Mc	Explorador de los archivos que trae consigo un editor llamado mcedit. El cual es necesario instalarlo.	sudo mc
cd nom_directorio	Sirve para cambiar de directorio.	cd sólo, va a tu home.
cd ..	Sube el nivel de los directorios.	Sube de nivel de /home/usuario/ pasa a /home/
cd -	Dirige al usuario a la ubicación donde se encontraba antes.	Si desde /home/usuario haces cd/etc/X1 y luego cd - vuelve a /home/usuario
cp archivo1 archivo2	Copia el archivo1 como un archivo2 (nuevo nombre) manteniendo siempre el original.	
cp archivo/directorio	Copia el archivo en el directorio manteniendo el nombre original.	
mkdir nom_directorio	Crea un nuevo directorio, nom_directorio.	
rmdir nom_directorio	Borra el directorio creado, nom_directorio.	
apt-getinstall nombre_paquete	Instala los paquetes.	apt-getinstallGnome
Chroot	Permite modificar el directorio principal. Sintaxis: chrootnom_directorio_raiz.	
Find	Busca los archivos que cumplen con la condición del directorio escogido.	findnom_directorio o nom_archivo condición.
Ifconfig	Sirve para obtener la información de la configuración de una red.	
Make	Esta herramienta controla la creación de los ejecutables y de otros archivos del programa comenzando por los archivos fuente.	make
Mkdir	Sirve para crear un nuevo directorio.	mkdir_nom_directorio

su o sudo	Este comando sirve para acceder al sistema como root. En Ubuntu se puede utilizar estos comandos como gksudo Kubuntu: kdesudo	
------------------	--	--

Tabla No. 3 Tabla de comandos

4. PROPUESTA.

4.1. TÍTULO DE LA PROPUESTA.

Diseño de algoritmos para la sincronización y codificación de canales de sistemas OFDM utilizando radios definidos por software.

4.2. JUSTIFICACIÓN.

Este capítulo describe los algoritmos propuestos para la Sincronización y Codificación de Canales de Sistemas OFDM Utilizando Radios Definidos Por Software.

Lo primero que se realizó fue la configuración de nuestro sistema operativo UBUNTU, instalando las siguientes librerías y aplicaciones, para la ejecución en GNU Radio:

- gr-recipes
- pybombs

4.3. DESCRIPCIÓN DE LA PROPUESTA.

Para la implementación de este proyecto se adquirirá el USRP B205mini-i para la realización de prácticas, con la ayuda de la herramienta de Software libre GNU Radio, se desarrollarán aplicaciones para un banco de pruebas con 6 algoritmos en OFDM.

Se propone realizar un banco de pruebas para poder simular la recepción, transmisión, de señales, con las cuales podremos analizar y estudiar la ortogonalidad de las señales con la sincronización y codificación de canales en sistemas OFDM.

Se analizarán los efectos del ruido, los niveles de recepción, la amplitud en que se transmiten y reciben las señales, las potencias de transmisión, recepción, así como también las representaciones de los diagramas de constelación que representan la relación de amplitud y fase de la portadora modulada digitalmente, para de esta manera práctica se llegue a la comprensión cabal de la modulación de las señales dentro del espectro electromagnético.

En este proyecto se presentan 6 algoritmos en OFDM para la sincronización y la estimación de canal de forma eficiente y robusta mediante el diseño ortogonal de preámbulos de tiempo, así como de frecuencia.

Se desarrollará un transmisor y un receptor para mostrar la aplicación de los algoritmos, utilizando GNU Radio y el Universal Software Radio Peripheral ETTUS.

La estructura que se trabajará, será bajo comunicaciones inalámbricas, tal como muestra la siguiente figura:

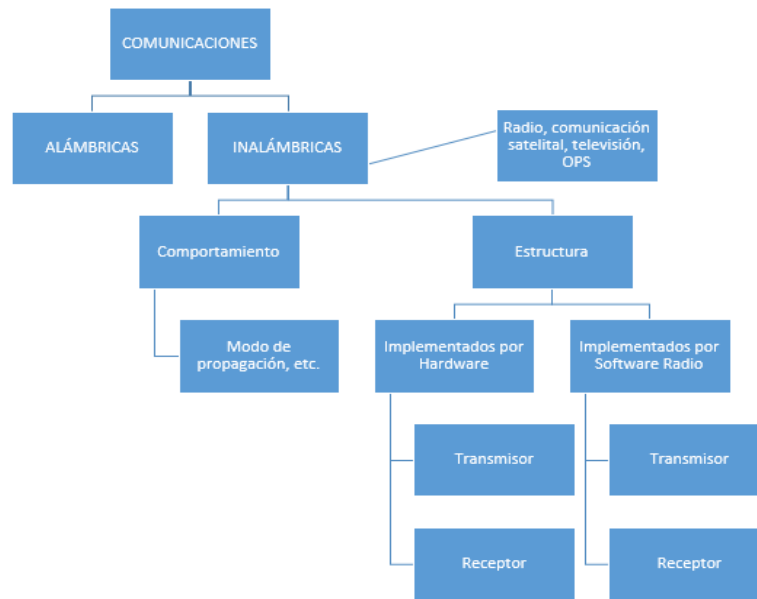


Figura 20 Mapa conceptual sobre comunicaciones inalámbricas.

Fuente: Software Radio y el USRP **Autor:** Jorge Alfonso Chávez Reyes

Para lograr implementar en software la parte de procesamiento de la señal se utilizarán sistemas de lógica programables como FPGA (Field Programmable Gate Array), DSP (Digital Signal Processor) y microprocesadores, ya que con se realiza el procesamiento digital de señales, debido a la implementación en software del procesamiento de la señal, es posible implementar diferentes tipos en el mismo sistema de radio.

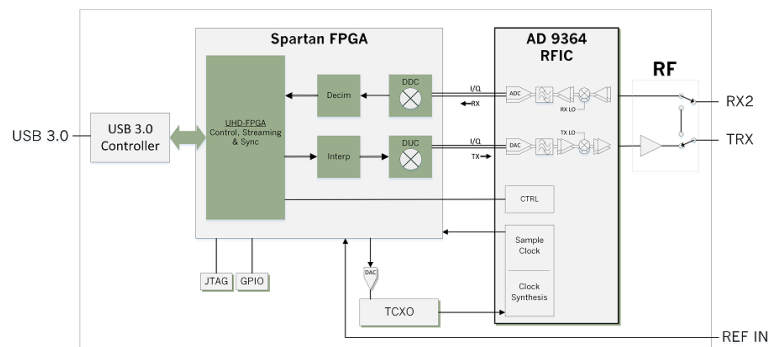


Figura 21 Arquitectura USRP B200mini Series.

Fuente: (Ettus, 2017)

Realizaremos mediante el software radio la modulación / demodulación, codificación / decodificación de las señales por medio de las puertas de arreglo en campos programables FPGA.

Se realizarán las gráficas en Python, el cual es un lenguaje de programación interpretado, interactivo y orientado a objetos.

En las prácticas se realizarán las siguientes configuraciones en el sistema GNU Radio, para que el mismo pueda servir en la realización de las prácticas de campo con las señales en el siguiente orden:

- **Práctica 1:** Instalación de Ubuntu en Máquina Virtual, instalación de GNU Radio y configuración del USRP.

Objetivos: Ilustrar el procedimiento para la creación de un disco virtual e instalar Ubuntu y así reconocer las conexiones del USRP.

- **Práctica 2:** Instalación de librerías adicionales.

Objetivos: Comprobar que la instalación de las librerías no genere errores durante dicho proceso y ejecutar pruebas de funcionamiento.

- **Práctica 3:** Análisis de la tasa de error binario (BER)

Objetivos: Configurar el equipo USRP B205mini-i, para la detección de una señal del entorno, ejecutar GNU Radio para la demostración de las diversas pérdidas de bits que pueden darse en la transmisión de una señal y demostrar el comportamiento de la señal mediante la interfaz gráfica de GNU Radio.

- **Práctica 4:** Codificación de canales.

Objetivos: Registrar los parámetros de configuración del equipo USRP y ejecutar GNU Radio para la observación gráfica de la frecuencia de señal de un canal determinado.

- **Práctica 5:** Radar.

Objetivo: Proporcionar un entorno genérico en GNU Radio para experimentar con varios tipos de radar.

- **Práctica 6:** Sincronización de canales.

Objetivos: Ingreso de parámetros del canal de transmisión para la recepción de una señal y analizar la ganancia y frecuencia del canal utilizado.

- **Práctica 7:** Transmisión y recepción OFDM

Objetivos: Construir bloques en GNU-Radio para el prefijo y la potencia cíclicos y asignar en bandas de subportadoras.

- **Práctica 8:** Wifi.

Objetivos: Estudiar el comportamiento de una señal Wifi, a través del procesamiento de señales en tiempo real para uso del sistema SDR.

5. RESULTADOS.

- Se diseñan algoritmos para realizar un banco de prueba necesario para la sincronización y codificación de canales de sistemas y tecnologías OFDM mediante USRP B205 mini i y GNU Radio implementado en el laboratorio de Telecomunicaciones de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana Sede Guayaquil.
- Se logran desarrollar seis algoritmos para un banco de pruebas en OFDM utilizando GNU Radio.
- Se programan aplicaciones para un receptor IEEE 802.11a/b/g/p, en el cual se realizará el análisis del tráfico que viaja por estas señales, así como también la forma en que se propaga la misma.
- El diseño de aplicaciones mediante codificación de canales para analizar su rendimiento en términos de tasa de error de bits (BER), para el mejoramiento en la eficiencia espectral es posible gracias a las prácticas empleadas.

5.1 ANÁLISIS DE RESULTADOS.

Práctica 1: Instalación de Ubuntu en Máquina Virtual, instalación de GNU Radio y configuración del USRP.

En la primera práctica se realiza la instalación del sistema operativo UBUNTU y la máquina virtual que permite la configuración del USRP B205mini-i. Durante este proceso de instalación y configuración de equipos, se realiza la descarga de librerías necesarias para el funcionamiento de la máquina virtual el GNU Radio.

Sin embargo, se instalaron más de 100 librerías en esta práctica, donde cada una de ellas cumplen una función específica, dependiendo de la modulación y tipo de práctica que se desea realizar en la máquina virtual. En conclusión, la VirtualBox emula una máquina virtual en el sistema operativo instalado en una máquina, por lo que no se afecta la configuración de ninguna forma. La conexión y configuración del USRP es sencilla de realizar siguiendo de manera ordenada los pasos para su instalación.

Práctica 2: Instalación de librerías adicionales.

En esta práctica se realiza la instalación de las diferentes librerías adicionales que necesita aplicación GNU Radio.

En conclusión:

✓ PyBombs es una herramienta para ayudar a instalar GNU Radio, módulos fuera de árbol y otros paquetes de software, de origen o binarios, en su mayoría independientes del sistema.

- ✓ PyBombs permite el uso de prefijos personalizados le permite desinstalar versiones más fácilmente y tener varias versiones instaladas una junto a la otra.

- ✓ PyBOMBS proporciona mucho valor ya que es fácil de instalar y está estrechamente integrado en el proyecto y la comunidad de GNU Radio.

Por lo cual se recomienda.

- ✓ Verificar las especificaciones técnicas para lograr un óptimo funcionamiento.

Práctica 3: Análisis de la tasa de error binario (BER).

En esta práctica se realiza el análisis de una tasa de error binario de una señal determinada en el entorno donde se ejecuta la aplicación GNU Radio. Para ello se configura el equipo USRP B205mini-i. Ver anexo 1.3

Luego de realizada la práctica se concluye lo siguiente:

- ✓ Se observa una BER del sistema OFDM con y sin estimación de canal y, en consecuencia, se dibuja un gráfico, que muestra que la tasa de errores de bits del sistema OFDM con estimación de canal es menor en comparación con uno sin estimación de canal.

- ✓ Se logró demostrar el comportamiento de las señales, mediante el análisis de la tasa de error binario, mediante el entorno gráfico de GNU Radio.

- ✓ Se evidencia visualmente el cambio en la frecuencia y la ganancia de la señal mediante el entorno gráfico de GNU Radio.

Además, se recomienda:

- ✓ Considerar los parámetros de potencia para evitar daños al equipo.

- ✓ Previo a la ejecución de la práctica, se debe tener instaladas las librerías instaladas.

- ✓ Se puede realizar investigación, utilizando diferentes esquemas de modulación, como QPSK, QAM, 8QAM, etc., en lugar de BPSK y elegir los mejores esquemas de sincronización y estimación de canal.

Práctica 4: Codificación de canales.

Esta práctica consiste en la configuración de los parámetros de codificación de una señal determinada en el ambiente, a través de GNU Radio.

En conclusión:

- ✓ La ventaja de utilizar radio definido por software es que se puede diseñar

y probar diferentes configuraciones en el menor tiempo posible.

✓ Se pueden desarrollar algunos bloques de procesamiento para mejorar la detección de símbolo en el receptor, los bloques con los que se trabajaron se configuraron a la necesidad del sistema emulado, sin embargo, esto abre una oportunidad para el desarrollo de propios bloques o la creación de una completa biblioteca para la simulación y emulación de sistemas de telecomunicaciones.

Por lo tanto, es recomendable:

✓ Incrementar las capacidades del equipo de cómputo a utilizar para la emulación, de un bus USB de 8MHz (2.0) a 32MHz (3.0) para utilizar un mayor ancho de banda en la transmisión.

✓ Mejorar las etapas de codificación y estimación de canal (completa implementación).

Práctica 5: Radar.

En la práctica cuatro, se refiere a la visualización de señales de radar mediante GNU Radio y USRP B205mini-i. Donde se ingresa la configuración de la señal con la cual se va a trabajar, para de esta manera visualizar gráficamente la señal en el programa.

La práctica concluye que:

✓ El receptor GND Radio OFDM puede servir como base para una mayor experimentación, mediciones e investigación en algoritmos de procesamiento de señales.

✓ Permite reproducibilidad de estudios conceptuales y experimentos, y todos los bloques del receptor se pueden analizar con más detalle mediante compañeros investigadores

Por consiguiente, también se recomienda:

✓ Cuando se investigan técnicas de SDR por primera vez, la radio GNU Se recomienda instalar el software y el controlador de hardware USRP en el Software VMware (o cualquier otro software similar) para configuraciones simples y modificaciones;

✓ Las computadoras más potentes con tarjetas Gigabit Ethernet deben usarse para mejorar el rendimiento del sistema de comunicación implementado;

✓ Se podría utilizar una frecuencia de referencia externa más precisa, como el GPS, para mejorar la precisión de la sincronización;

- ✓ Podrían utilizarse atenuadores de potencia para resolver los problemas tempranos de saturación de potencia en los receptores, por lo que se podría ampliar el rango experimental de SNR.

Práctica 6: Sincronización de canales.

Al igual que en las prácticas anteriores, se utiliza una configuración del sistema GNU Radio, para la sincronización de canales para la recepción de una señal determinada. Ver anexo 1.6

La práctica establece que:

- ✓ La técnica OFDM se usa ampliamente en aplicaciones en tiempo real como los sistemas de evaluación a largo plazo (LTE), WiFi, WiMax.
- ✓ La implementación de comunicaciones inalámbricas utilizando software definido por radio (SDR) es de código abierto y una de las áreas de investigación emergentes. SDR es una técnica para aplicaciones inalámbricas de alta velocidad con dispositivos de hardware mínimos y también rentables.
- ✓ En este documento, hemos implementado un sistema de comunicación basado en OFDM con modelo de canal, estimación de canal y sincronización utilizando GNU Radio, que es una fuente abierta.

Además, se recomienda:

- ✓ Incrementar las capacidades del equipo de cómputo a utilizar para la emulación, de un bus USB de 8MHz (2.0) a 32MHz (3.0) para utilizar un mayor ancho de banda en la transmisión.
- ✓ Mejorar las etapas de codificación y estimación de canal (completa implementación).

Práctica 7: Transmisión y recepción OFDM.

Esta práctica se compone de dos partes, transmisión en OFDM y recepción en el mismo sistema. Para ello también se realiza la configuración del sistema y equipos a utilizar para que se pueda visualizar los respectivos gráficos.

En esta práctica se concluye lo siguiente:

- ✓ La naturaleza "ortogonal" de la multiplexación de frecuencia, seguida del prefijo elimina efectivamente ISI en canales de ruta única y lo reduce considerablemente en canales de ruta múltiple.
- ✓ OFDM guarda el espectro en comparación con la multiplexación por división de frecuencia regular, debido a la superposición, pero a las

portadoras ortogonales que comparten el mismo espectro.

✓ El esquema de prefijo cíclico y DFT permiten una señal rápida y fácil procesamiento, esto sucede en el hardware y en software, lo que permite una fácil estimación y ecualización del canal, a través de símbolos fijos conocidos (símbolos piloto).

✓ OFDM puede admitir el acceso dinámico a paquetes. La codificación espacio-temporal pueden realizarse en OFDM y todos los beneficios se pueden obtener fácilmente. Modulación adaptativa y tono/potencia.

Finalmente se recomienda:

✓ OFDM es altamente sensible a los errores de sincronización de tiempo y frecuencia, especialmente con los errores de sincronización de frecuencia.

✓ La demodulación de una señal OFDM con un desplazamiento en la frecuencia puede conducir a una alta tasa de error de bits.

Práctica 8: Wifi.

Finalmente, la presente práctica se refiere al análisis de señales wifi en el entorno donde se realiza la práctica del sistema GNU Radio y el USRP B205mini-i.

Por lo tanto, se ingresan las frecuencias en las cuales se va a trabajar para receptar la señal Wifi y analizarla.

En conclusión:

✓ En esta práctica se demuestra el desarrollo de la radio definida por software (SDR) que utiliza el banco de pruebas SDR04 y el kit de herramientas complementario de GNU Radio. IEEE802.11a OFDM se ejecuta con codificación de canal considerando el codificador FEC. IEEE 802.11a determinaciones de transceptor OFDM en diferentes esquemas de modulación, niveles de ruido del canal y variación de fase.

✓ En el estudio comparativo, se encontró que: cuando las condiciones del canal son deficientes, se usaron esquemas de eficiencia energética como BPSK y, a medida que la calidad del canal mejora, se usó QPSK.

Se recomienda para ello:

Ajustar el método de modulación para una transferencia de datos óptima, lo cual aumenta la eficiencia del ancho de banda y aumenta la capacidad general del sistema.

6. CONCLUSIONES.

- A través del uso de GNU Radio se desarrollan seis algoritmos para un banco de pruebas en OFDM.
- La captación de las señales es posible gracias a la modulación y demodulación de OFDM en las tecnologías más conocidas, simulando e implementando en un SDR, el espectro OFDM la recepción y transmisión de paquetes OFDM utilizando módulos USRP B205 mini-i.
- Se diseñan y programan aplicaciones para un receptor IEEE 802.11a/b/g/p, en el cual se realiza el análisis del tráfico que viaja por estas señales, así como también la forma en que se propaga la misma.
- Para analizar su rendimiento en términos de tasa de error de bits (BER), se diseñan aplicaciones mediante codificación de canales.

7. RECOMENDACIONES.

- Utilizar un computador con procesador I7 para garantizar el correcto funcionamiento del software y captación de señal.
- Conectar de forma óptima los equipos de manera adecuada siguiendo las especificaciones técnicas del manual de usuario.
- Asignar la potencia adecuada al equipo según las especificaciones para evitar daños permanentes al mismo durante la transmisión de señal.
- Instalar las librerías necesarias para la correcta ejecución del programa utilizado en las prácticas.

8. PRESUPUESTO.

DESCRIPCIÓN	CANT.	PRECIO	SUBTOTAL
USRP B205MINI-I	1	\$ 1,200.00	\$ 1,200.00
Capacitación		\$ 1,000.00	\$ 1,000.00
		TOTAL	\$ 2,200.00

Tabla No. 4 Presupuesto para el proyecto de titulación

9. CRONOGRAMA.

ACTIVIDAD	FECHA													
	2017				2018									
	Mayo - Junio	Ago - Sept	Oct - Nov	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre
Compra de equipos														
Recolección de la información														
Investigación y aprendizaje de los USRP														
Diseño de los algoritmos														
Diseño de diagramas de bloques para control del programa														
Práctica 1														
Práctica 2														
Práctica 3														
Práctica 4														
Práctica 5														
Práctica 6														
Práctica 7														
Práctica 8														
Diseño de módulos extras y auxiliares														
Fase de prueba														
Corrección de errores														

Tabla No. 5 Cronograma

REFERENCIAS BIBLIOGRÁFICAS.

- Andreotti, J. I. (2015). *¿Qué es el BER (Bit Error Rate)?* Buenos Aires - Argentina: Magna Editores.
- Arias, F. (2013). *El proyecto de investigación. Introducción a la metodología científica*. Caracas: Editorial Episteme, C.A.
- Arraño, H., & Azurdia, C. (2015). *OFDM: Hoy y en el futuro de las comunicaciones*. Santiago de Chile: XV Congreso Internacional de Telecomunicaciones SENACITEL.
- Cadena Iñiguez, P. (2017). Métodos cuantitativos, métodos cualitativos o su combinación en la investigación: un acercamiento en las ciencias sociales. *Revista Mexicana de Ciencias Agrícolas*, 1603-1617.
- Carlson, A. B., Crilly, P. B., & Rutledge, J. C. (2015). *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. New York: McGraw-Hill Higher Education,.
- Escobar, R. (2016). *Core concepts of GNU Radio*. Retrieved from <http://stg-pepper.blogspot.com/2016/12/iniciando-con-gnu-radio.html>
- Ettus. (2017). *Ettus Research*. Retrieved from https://www.ettus.com/content/images/B205mini-i_Top-Large_1.jpg
- Facal, T. (2015). *GUía para elaborar un proyecto de investigación social*. Madrid: Ediciones Paraninfo, S.A.
- Fernández Díaz, J. (2015). *Telecomunicaciones digitales, dispositivos y sistemas*. Nuevo León - México: Universidad Autónoma de Nuevo León.
- López Cortez, S. (2015). *Análisis matemático del desempeño de OFDM en función del tiempo de guarda en canales con desvanecimiento selectivo en frecuencia*. Sangolquí - Ecuador: Universidad de las Fuerzas Armadas.
- López, E., Zinemanas, P., & Ramos, M. (2016). *Introducción a los sistemas de comunicación*. Rocha - Uruguay: Centro Universitario Regional Este.
- López, V. (2016). *Problemas resueltos de electromagnetismo*. Madrid: Editorial Centro de Estudios Ramon Areces S.A.
- Marcano, M. (2018). *Investigación experimental*. Maracaibo - Venezuela: Universidad Fermín Toro.
- Masero, V. (2015). *Principios de OFDM*. Sevilla - España: Universidad de Sevilla.
- Muñoz, J. T. (2018). *Comunicaciones digitales 2*. Málaga - España:

RIUMA.

Prieto Del Amo, C. (2015). *Estimación de canal y desplazamiento de frecuencia en sistemas MIMO-OFDM con prefijo cíclico insuficiente*. Madrid - España: Universidad Carlos III de Madrid.

Quesada, P. (2016). *Desarrollo de un sistema de transmisión (emisor-receptor) con módulos bluetooth*. Madrid - España: Universidad Politécnica de Madrid.

Rodríguez, E. (2013). *Metodología de la Investigación*. Villahermosa: Universidad de Juárez Autónoma de Tabasco.

Suárez, A. (2015). *Sistemas de comunicación digital*. Nuevo León - México: Universidad Autónoma Nuevo León.

Universia Costa Rica. (2017, septiembre 04). *Tipos de investigación: Descriptiva, Exploratoria y Explicativa*. Retrieved from Universia Costa Rica:

<http://noticias.universia.cr/educacion/noticia/2017/09/04/1155475/tipos-investigacion-descriptiva-exploratoria-explicativa.html>

Uruchima Portilla, A. A. (2016). *Análisis del desempeño de sistemas OFDM con diferentes modulaciones usando Simulink*. Guayaquil - Ecuador: Universidad Católica de Santiago de Guayaquil.

Viveros, J. (2015). *Principios de comunicaciones digitales*. México D.F.: Universidad Autónoma Metropolitana.

GLOSARIO.

OFDM: La Multiplexación por División de Frecuencias Ortogonales, en inglés Orthogonal Frequency Division Multiplexing (OFDM), o Discrete Multi-tone Modulation (DMT) Esta es una técnica de transmisión que consiste en la multiplexación de conjuntos de portadoras de diferentes frecuencias, cada una de las cuales está modulada con QAM o PSK. (Esacademic, 2018)

GNU RADIO: Es un sistema operativo de tipo Unix desarrollado por y para el Proyecto GNU, y auspiciado por la Free Software Foundation. Está formado en su totalidad por software libre, mayoritariamente bajo términos de copyleft.

USRP: Programador Periférico Universal de Radio, es un transceptor flexible y rentable que convierte una PC estándar en un potente sistema de generación de prototipos inalámbricos. (Instruments, 2017)

SSID: (Service Set Identifier) SSID es el nombre que identifica de la LAN inalámbrica. Una cadena de texto que distingue entre mayúsculas y minúsculas que acepta hasta 32 caracteres alfanuméricos y se utiliza en el proceso de unirse a la red inalámbrica.

ADC: Un conversor o convertidor de señal analógica.

DSSS: Del inglés Direct Sequence Spread Spectrum es el método para la codificación de canal (antes de la modulación) en el espectro ampliado para realizar la transmisión de las señales digitales sobre las ondas radiofónicas más utilizadas.

FPGA: Una matriz de puertas programables o FPGA (del inglés Field Programmable Gate Array), es un dispositivo que puede programarse y que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.

DSP: Un procesador digital de señales o DSP (Digital Signal Processor) es un sistema basado en un procesador o microprocesador que posee un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesamiento y representación de señales analógicas en tiempo real.

SDR: Radio definida por software o SDR (Software defined radio) es un sistema de radiocomunicaciones donde varios componentes típicamente implementados en hardware (mezcladores, filtros, moduladores/demoduladores, detectores, etc) son implementados en software, utilizando un ordenador personal u otros dispositivos de computación embebidos. Aunque el concepto SDR no es nuevo, la reciente evolución de la circuitería digital ha hecho posible desde el punto de vista práctico muchos de los procesos que tiempo atrás eran solamente posibles desde un punto de vista teórico.

UHD: Ultra Hi-Definition (en español ultra alta definición), también conocido como UHD TV (Ultra High Definition Television), se refiere a un formato de vídeo digital, será usado para pantallas con una relación de aspecto de por lo menos 16:9 y al menos un puerto digital capaz de portar video nativo a una resolución mínima de 3840x2160 píxeles.

FFT: La transformada rápida de Fourier, conocida por la breviatura FFT (Fast Fourier Transform) es un algoritmo eficiente que permite calcular la transformada de Fourier discreta (DFT) y su inversa. La FFT es de gran importancia en una amplia variedad de aplicaciones, desde el tratamiento digital de señales y filtrado digital en general a la resolución de ecuaciones en derivadas parciales o los algoritmos de multiplicación rápida de grandes enteros.

LTE: De las siglas Long Term Evolution (evolución a largo plazo) y hace referencia a la tecnología de banda ancha inalámbrica que sirve para la

transmisión de datos con la finalidad de dar acceso a Internet a los dispositivos móviles.

TDТ: Televisión digital terrestre, también llamada en algunos países de América televisión digital abierta (TDA), es la transmisión de imágenes en movimiento y su sonido asociado mediante codificación binaria a través de una red de repetidores terrestres.

GSM: El sistema global para las comunicaciones móviles (del inglés Global System for Mobile communications, GSM, y originariamente del francés groupe special mobile) es un sistema estándar, libre de regalías, de telefonía móvil digital.

BER: La tasa de error binario o Bit Error Ratio, se define como el número de bits recibidos de forma incorrecta respecto al total de bits enviados durante un intervalo específico de tiempo. Es usado en telecomunicaciones para modelar un canal de comunicación.

DCS: Sistema de Control Distribuido o DCS, es un Sistema de control aplicado a procesos industriales complejos en las grandes industrias como petroquímicas, papeleras, metalúrgicas, centrales de generación, plantas de tratamiento de aguas, incineradoras o la industria farmacéutica.

TDM: Multiplexación por división en el tiempo (TDM Time-Division Multiplexing), sincroniza los datos de cada una de las fuentes y son divididos en ranuras temporales. Se van enviando a una ranura de cada fuente a continuación de la ranura anterior, de forma cíclica. El resultado final es un conjunto de bits en el canal que pertenecen a distintas comunicaciones. Este tipo de multiplexación se utiliza para el envío de voz digitalizada.

DFT: Transformada Discreta de Fourier (discrete Fourier transform) es un tipo de transformador discreto utilizada en el análisis de Fourier.

IFFT: Inverse Fast Fourier transform. Es un algoritmo para realizar la transformada de Fourier inversa, que deshace el proceso de DFT e IDFT de una secuencia {}.

FEC: La corrección de errores hacia adelante (Forward Error Correction) consiste en un tipo de mecanismo de corrección de errores que permite su rectificación en el receptor sin producir retransmisión de la información original. Se utiliza en sistemas en tiempo real donde no se puede esperar a la retransmisión para mostrar los datos o en aquellos sistemas sin retorno.

xDSL: Del inglés X Digital Subscriber Line, tecnología que soporta un gran ancho de banda con unos costos de inversión relativamente bajos y que trabaja sobre la red telefónica ya existente.

BPSK: Bi Phase-Shift Keying Modulación de desplazamiento de fase de 2

símbolos. Emplea 2 símbolos, con 1 bit de información cada uno.

QPSK: Quadrature Phase-Shift Keying, es conocido como PSK Cuaternaria, esta modulación digital de constelación por cuatro puntos equidistantes del origen de coordenadas. Con cuatro fases, qPSK puede codificar dos bits por cada símbolo.

QAM: Modulación de amplitud en cuadratura (Quadrature Amplitude Modulation) es una técnica que transporta dos señales independientes, mediante la modulación, tanto en amplitud como en fase, de una señal portadora. Esto se consigue modulando una misma portadora, desfasada en 90° . La señal modulada en QAM está compuesta por la suma lineal de dos señales previamente moduladas en doble banda lateral con portadora suprimida.

ISI: Interferencia entre símbolos, es una forma de distorsión de una señal en la cual un símbolo interfiere con símbolos posteriores. Es un fenómeno no deseado ya que los símbolos anteriores tienen un efecto similar al del ruido, lo que hace que la comunicación sea menos fiable. La extensión/propagación del pulso más allá del intervalo de tiempo asignado hace que interfiera con los pulsos/ritmos vecinos. ISI es normalmente causada por la propagación por trayectos múltiples o respuesta de frecuencia lineal o no lineal inherente de un canal que hace que los símbolos sucesivos se desenfocuen juntos.

ICI: Interferencia entre canales.

CP: Prefijo cíclico, intervalo de tiempo, con esto se pueden concatenar varios OFDM uno detrás de otro sin miedo a que el eco del medio de propagación los mezcle entre sí.

SNR: Relación señal/ruido (signal to noise ratio) se define como la proporción existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe.

AWGN: Ruido Gaussiano Blanco Aditivo (Additive White Gaussian noise) es un modelo de ruido básico utilizado en la teoría de la información para imitar el efecto de muchos procesos aleatorios que ocurren en la naturaleza. Los modificadores denotan características específicas:

Aditivo porque se agrega a cualquier ruido que pueda ser intrínseco al sistema de información.

Blanco se refiere a la idea de que tiene un poder uniforme en la banda de frecuencia para el sistema de información. Es una analogía con el color blanco que tiene emisiones uniformes en todas las frecuencias en el espectro visible. Gaussiano porque tiene una distribución normal en el dominio de tiempo con un valor de dominio de tiempo promedio de cero.

CFO: Desplazamiento de frecuencia más corto.

GRC: Es un entorno de desarrollo que facilita la construcción de aplicaciones para GNU RADIO sin la necesidad estricta de saber programar en Python o C++, dado que GRC produce justamente el código Python correspondiente a la aplicación construida gráficamente. Facilita también la corrección de errores y las pruebas –ejecución- de la aplicación que se construye.

SWIG: (Simplified Wrapper and Interface Generator) Es una herramienta de desarrollo de software que conecta programas escritos en C y C++ con una variedad de lenguajes de programación de alto nivel. SWIG se utiliza con diferentes tipos de idiomas de destino, incluidos los lenguajes de script comunes como Javascript, Perl, PHP, Python, TCL y Ruby.


IEEE: (Institute of Electrical and Electronics Engineers) Instituto de Ingeniería eléctrica y electrónica es una asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas.

MHz: Megahercio es una medida de la frecuencia, equivale a 10^6 hercios (1 millón). Se utiliza muy frecuentemente como unidad de medida de la frecuencia de trabajo de un dispositivo o como medida de ondas.

ANEXOS.

GUÍAS DE PRÁCTICAS DE LABORATORIO.

PRÁCTICA 1.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Instalación de Ubuntu en Máquina Virtual, instalación de GNU Radio y configuración del USRP
OBJETIVO GENERAL: Instalar Ubuntu utilizando máquina virtual y configurar la conexión del USRP. OBJETIVOS ESPECÍFICOS: ➤ Ilustrar el procedimiento para la creación de un disco virtual e instalar Ubuntu. ➤ Reconocer las conexiones del USRP.		
INSTRUCCIONES		
Introducción Máquina Virtual Una máquina virtual es un software que permite que sobre un sistema operativo se ejecute otro sistema sin necesidad de modificar el sistema operativo instalado en nuestra máquina. Ubuntu Es una distribución de Linux es decir es un sistema operativo libre y más fácil de usar que otras distribuciones Linux. Al ser libre cualquier persona puede usarlo para cualquier trabajo sin necesidad de dar aviso al desarrollador o entidad.		

NI USRP

Es una plataforma de radios definidos por software para la generación de varios prototipos de comunicación. Es usada para la investigación y en prácticas de telecomunicaciones. Sus aplicaciones son la modulación analógica y modulación digital entre las que sobresalen AM, FM, FSK, PSK, QAM, etc., también su uso con GPS y OFDM. NI USRP es un transceptor, es decir que por aire o por cable puede recibir y transmitir señales.

Actividades por desarrollar

PARTE 1

Instalar la máquina virtual.

Paso 1

La máquina virtual a usarse en la práctica es Oracle VM VirtualBox.

Ruta de descarga:

<https://www.virtualbox.org/wiki/Downloads>

Paso 2

Luego de la instalación de Oracle VM VirtualBox procedemos a abrir el programa y se muestra la siguiente pantalla:

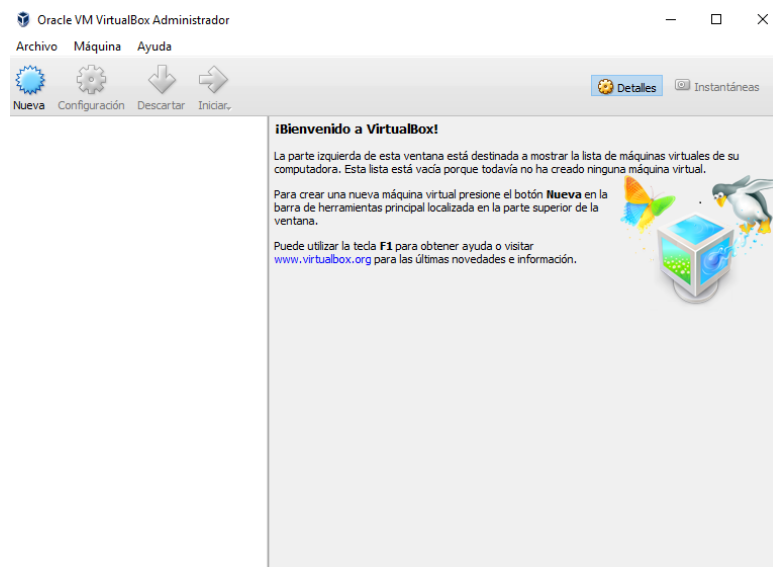


Figura 22 Pantalla Principal de Oracle VM VirtualBox
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Crear una máquina virtual

Damos clic en el icono Nueva, luego se coloca el Nombre a la máquina virtual en este caso “Ubuntu”, también se elige el Tipo “Linux” y la Versión “Ubuntu 64 bits”.

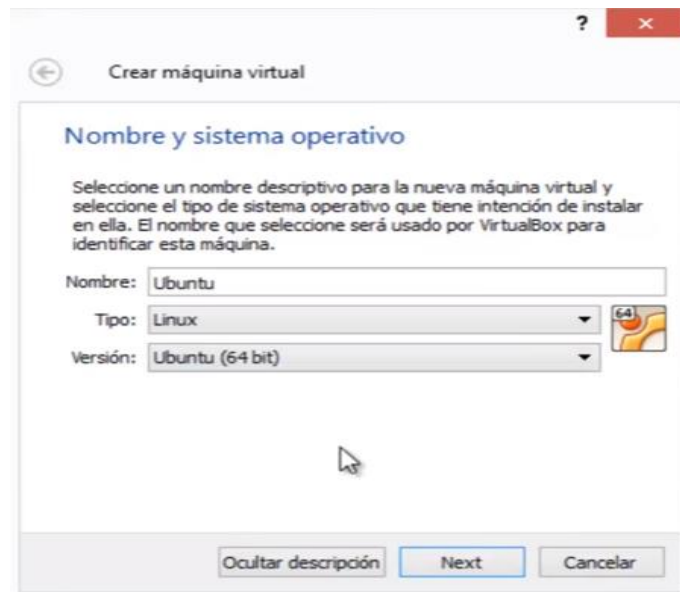


Figura 23 Crear máquina virtual.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 4

Asignar la memoria RAM a la máquina virtual, se recomienda usar 2Gb.

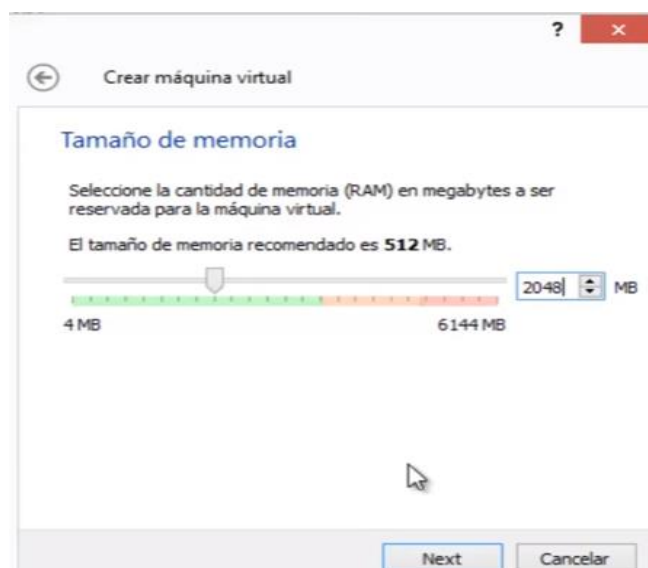


Figura 24 Tamaño de la memoria RAM de la máquina virtual.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 5

Seleccionar la opción “Crear un disco virtual ahora” de tamaño recomendado 8Gb.

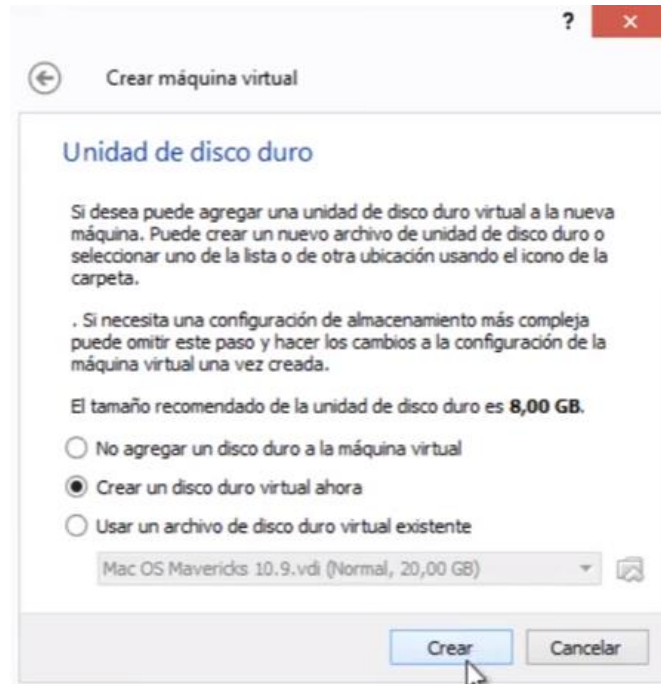


Figura 25 Creación del disco duro virtual.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 6

Seleccionar la opción “VDI (VirtualBox Disk Image)”.

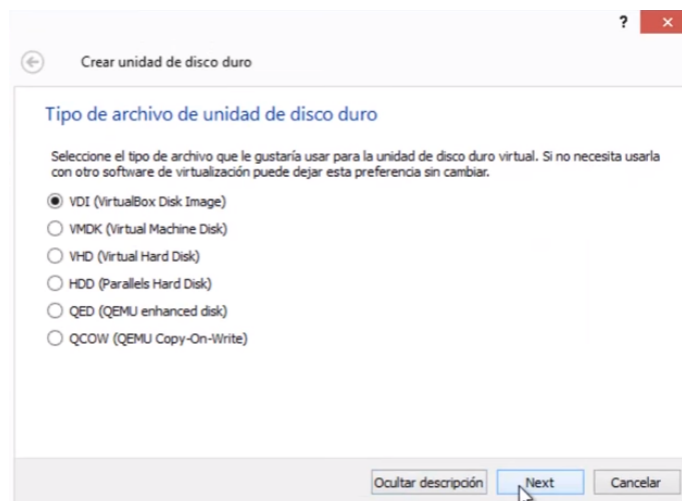


Figura 26 Tipo de archivo de disco duro

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 7

Escoger la opción “Reservado dinámicamente”, esto para que ocupe el espacio que realmente sea necesario.

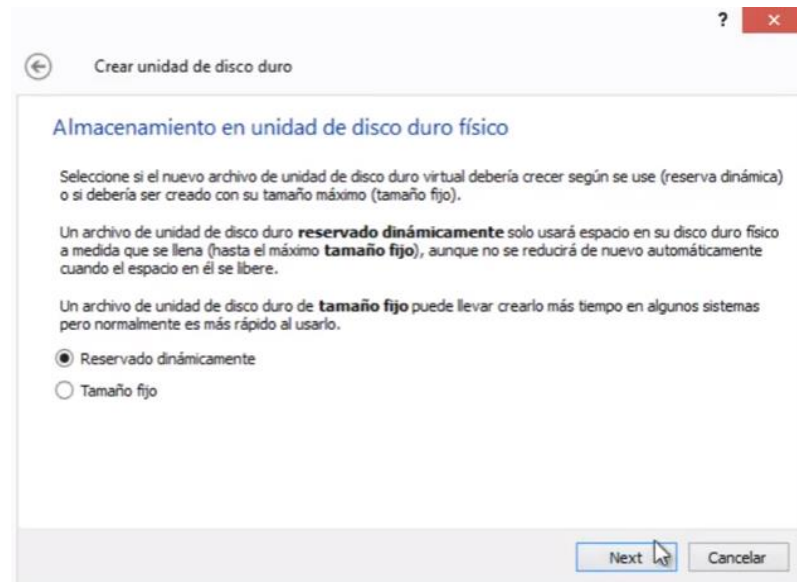


Figura 27 Tamaño del disco duro.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 8

Asignar la ubicación del archivo y el tamaño del disco duro, luego colocar el nombre de la carpeta y seleccionar el tamaño del disco duro.

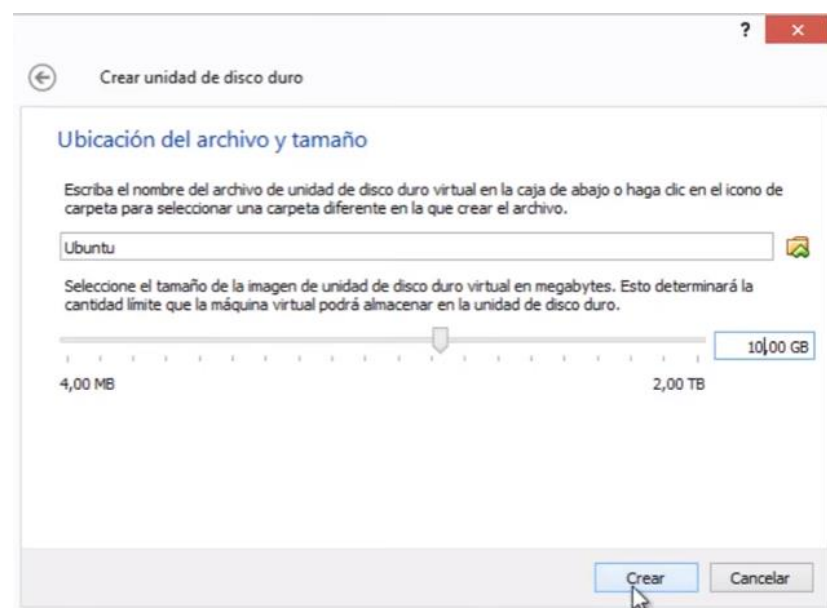


Figura 28 Ubicación del archivo y tamaño.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

PARTE 2

Instalar Ubuntu

Paso 1

Dando clic derecho, abrir la configuración de la máquina virtual.

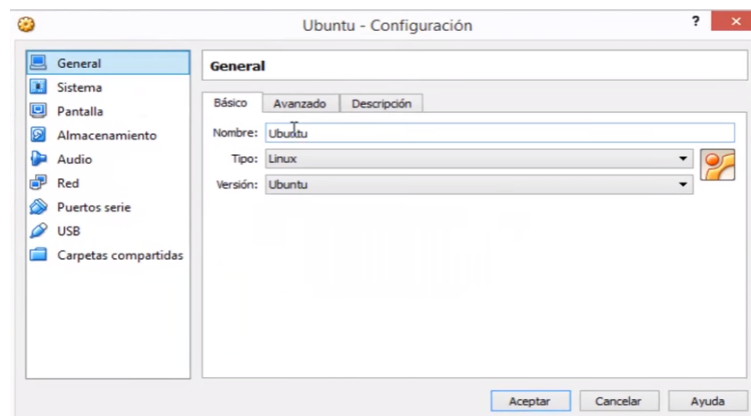


Figura 29 Propiedades de la máquina virtual.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 2

Escoger la opción “Almacenamiento” y seleccionamos “Controlador: IDE”

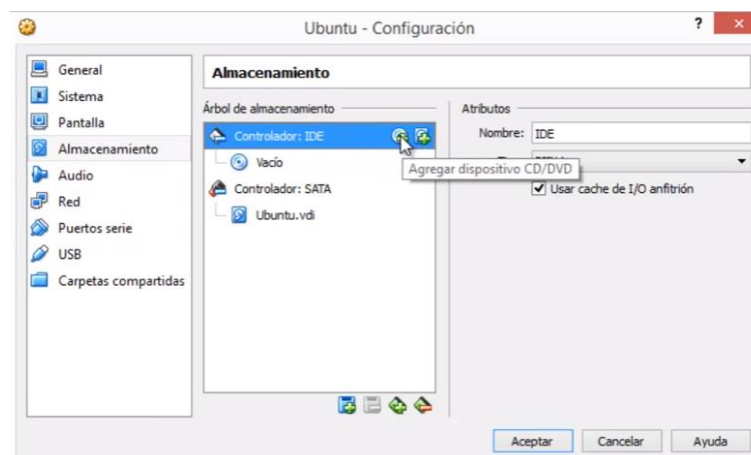


Figura 30 Agregar dispositivo.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Seleccionamos el disco, en este caso el Ubuntu que ya ha sido descargado

con anterioridad.

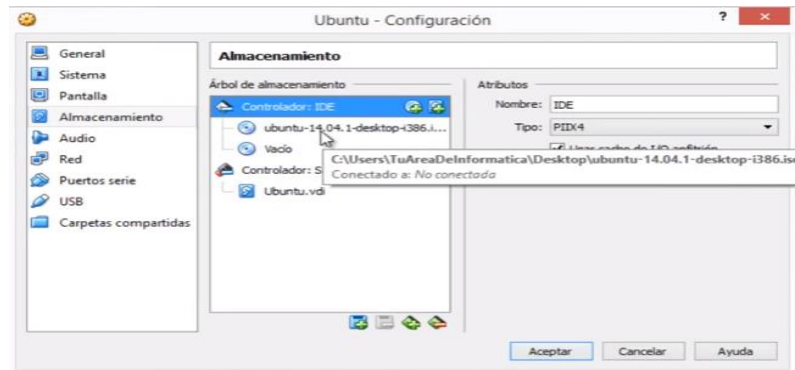


Figura 31 Cargar imagen del Ubuntu.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 4

Iniciar la máquina virtual, se nos presenta la siguiente pantalla, escogemos el idioma y luego le damos clic en “Instalar Ubuntu”.



Figura 32 Instalar Ubuntu.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 5

Damos clic en “Instalar ahora”.

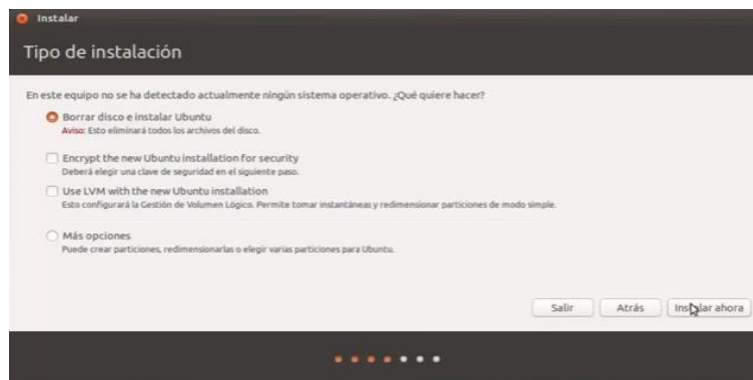


Figura 33 Tipo de instalación.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 6

Configurar la zona horaria, en este caso Guayaquil, luego dar clic en continuar.



Figura 34 Configuración de zona horaria.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 7

Configurar el teclado, escoger la opción “español (latinoamericano)”.

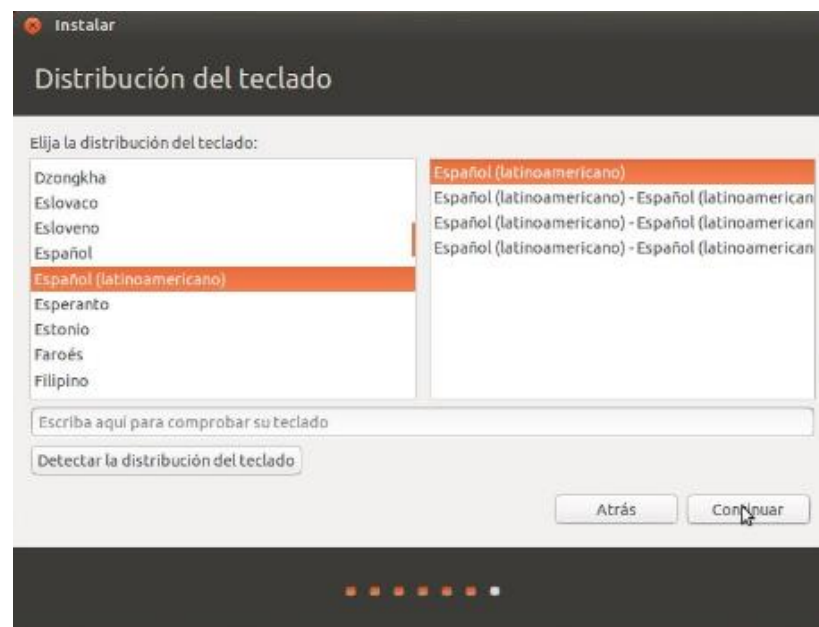


Figura 35 Configuración del teclado.

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 8

Asignar un nombre y contraseña para ingresar a la máquina virtual, luego

dar clic en continuar.

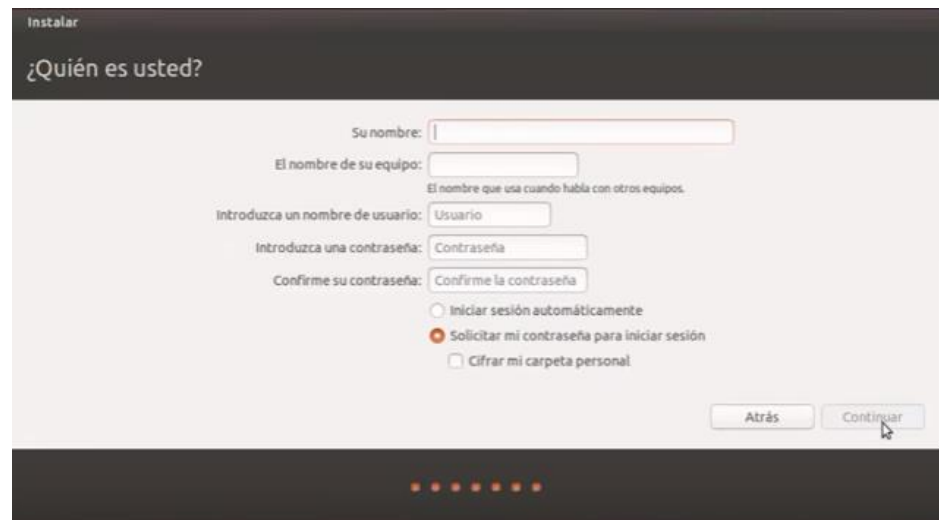


Figura 36 Asignar nombre y contraseña a la máquina virtual.
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 9

Reiniciar el sistema y ya tendremos Ubuntu funcionando.



Figura 37 Escritorio de Ubuntu.
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

PARTE 3

Instalar GNU Radio y configurar el USRP

Paso 1

Instalación del driver del USRP en Windows

Vamos a la página siguiente:

https://files.ettus.com/manual/page_install.html

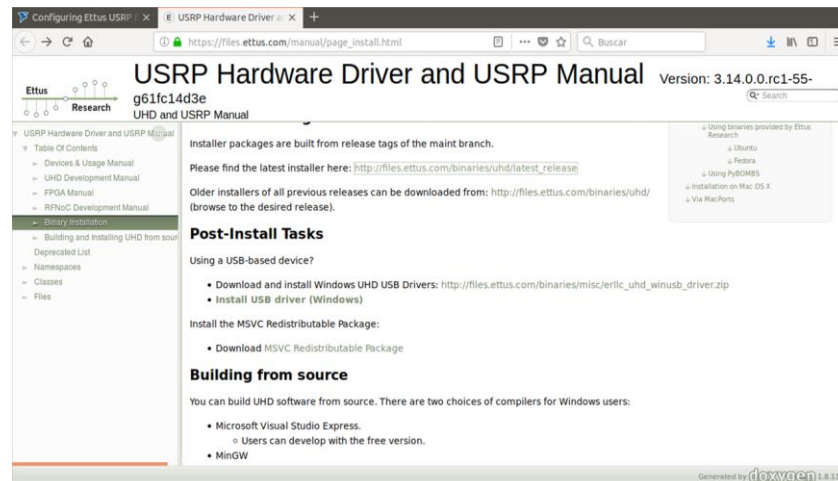


Figura 38 Instalación del driver del USRP en Windows
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Seleccionamos Install USB driver (windows)

Aparecerá la siguiente ventana donde seleccionamos

Download the driver from the UHD wiki page [here](#).

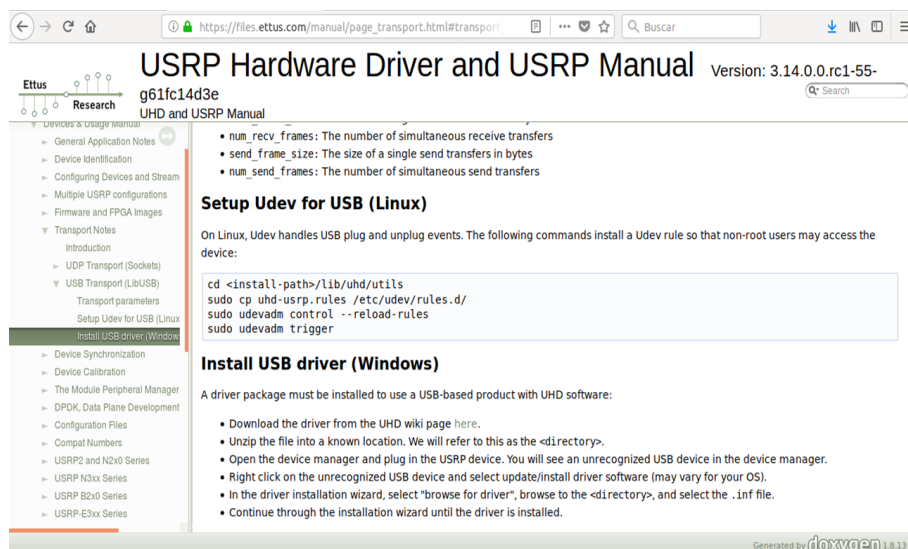


Figura 39 Instalación USB driver
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Damos clic y nos aparecerá lo siguiente

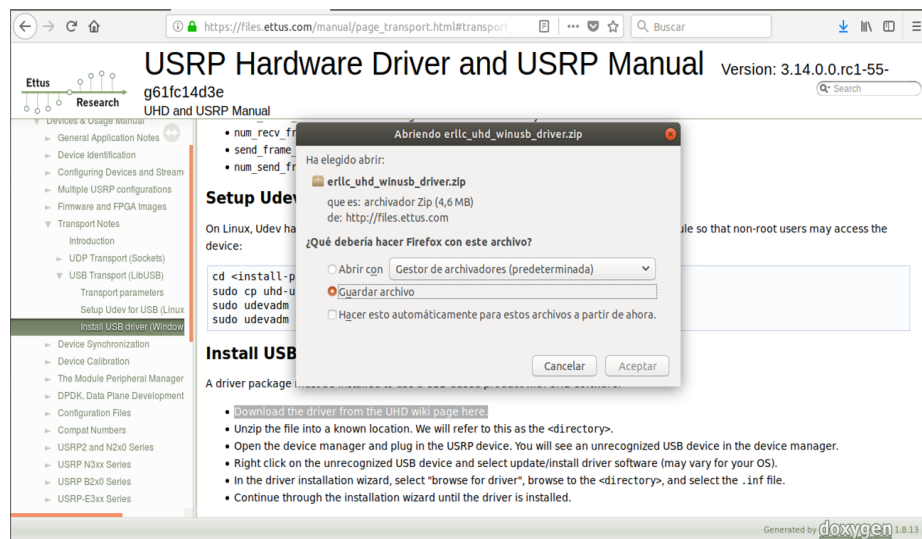


Figura 40 Abriendo USB driver

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Descargamos y descomprimos el archivo.

Nos vamos al Administrador de dispositivos y buscamos el dispositivo que esté con la interrogante.

Actualizamos el driver del dispositivo con los drivers previamente descargados.

Abrir la terminal de Ubuntu y escribir los siguientes códigos para descargar el driver y la librería de GNU Radio.

Driver: `sudo apt-get install mosmo-sdr`

Librería: `sudo apt-get install gr-osmosdr`

Paso 2

Como se muestra en la figura:

Conecte la antena al USRP en la terminal TRX.

Conecte el cable USB 3.0 en la terminal USB3.



Figura 41 USRP B205mini-i

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Proceder a cambiar la dirección IP de su computadora:

Dar clic en “Editar las conexiones”.

Escoger la conexión cableada que esta conectad y dar clic en “Editar”.

Dar clic en “Ajustes de IPv4”, en Método escoger la opción “Manual”, después proceder a cambiar la Dirección y la Máscara de red, y por último le damos clic en Guardar.

Paso 4

Instalar el driver de nuestro USRP

```
sudo add-apt-repository ppa:ettusresearch/uhd
```

```
sudo apt-get update
```

```
sudo apt-get install libuhd-dev libuhd003 uhd-host
```

Paso 5

Para comprobar que el USRP está conectado en su computador, ingrese a la terminal y escribir el siguiente comando: `uhd_find_devices`

CONCLUSIONES

- ✓ VirtualBox emula una máquina virtual en el sistema operativo instalado en una máquina, por lo que no se afecta la configuración de ninguna forma.
- ✓ La conexión y configuración del USRP es sencilla de realizar siguiendo de manera ordenada los pasos para su instalación.

RECOMENDACIONES


Es recomendable tener un procesador aceptable, como mínimo 1Gb de RAM y 10 Gb libres de disco duro

Se recomienda que la imagen ISO de Ubuntu sea desmontada una vez instalado el sistema en la máquina virtual.

Recordar cambiar la IP a Manual y colocar la IP propia del USRP.

Al final de cambiar la IP no olvidar comprobar si el USRP está conectado a su computadora.

PRÁCTICA 2.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Instalación de librerías adicionales
OBJETIVO GENERAL: Realizar la instalación de las librerías que sean necesarias para el correcto funcionamiento del software GNU Radio OBJETIVOS ESPECIFICOS: ➤ Comprobar que la instalación de las librerías no genere errores durante dicho proceso. ➤ Ejecutar pruebas de funcionamiento.		
INSTRUCCIONES		
1. Introducción GNU Radio es un conjunto de herramientas de desarrollo de software de fuente abierta y gratuita que proporciona bloques de procesamiento de señales para implementar radios de software. Se puede usar con hardware de RF externo de bajo costo y fácilmente disponible para crear radios definidos por software, o sin hardware en un entorno similar a la simulación. Se usa ampliamente en entornos de aficionados, académicos y comerciales para apoyar la investigación de comunicaciones inalámbricas y los sistemas de radio del mundo real. 2. Actividades por desarrollar Paso 1 PyBOMBS PyBOMBS (Python Build Overlay Managed Bundle System) es el método recomendado para construir e instalar GNU Radio. Consulte		

<https://github.com/gnuradio/pybombs> para obtener instrucciones detalladas. Las instrucciones abreviadas se duplican a continuación.

1.- Instalar PyBOMBS:

```
$ [sudo] pip install PyBOMBS
```

o

```
$ git clone https://github.com/gnuradio/pybombs.git
```

```
$ cd pybombs
```

```
$ sudo python setup.py install
```

2.- Añadir librerías PyBOMBS:

```
$ pybombs recipes add gr-recipes git+https://github.com/gnuradio/gr-recipes.git
```

```
$ pybombs recipes add gr-etcetera git+https://github.com/gnuradio/gr-etcetera.git
```

3.- Configure un prefijo de instalación:

```
$ pybombs prefix init ~/prefix/default/
```

4.- Instala GNU Radio:

```
$ pybombs install gnuradio
```

5.- Ejecute GNU Radio Companion desde su nuevo prefijo:

```
$ source ~/prefix/default/setup_env.sh
```

```
$ gnuradio-companion
```

O ejecutarlo sin cambiar el entorno actual.

```
$ pybombs run gnuradio-companion
```

Paso 2

Instalación de repositorio

Fuente de origen manual

Las instrucciones completas de compilación se detallan en la Guía de compilación de radio de GNU. Las instrucciones abreviadas se duplican a continuación. Asegúrese de que ha satisfecho las dependencias externas,

consulte Dependencias de radio GNU.

1.- Echa un vistazo al último código:

```
$ git clone https://github.com/gnuradio/gnuradio.git
```

2.- Construir con CMake:

```
$ cd gnuradio
```

```
$ mkdir build
```

```
$ cd build
```

```
$ cmake [OPTIONS] ../
```

```
$ make
```

```
$ make test
```

```
$ sudo make install
```

[OPTIONS] incluye establecer el prefijo de instalación -DCMAKE_INSTALL_PREFIX=<directory to install to> y el tipo de compilación -DCMAKE_BUILD_TYPE=<type>. Actualmente, GNU Radio tiene un "Debug" tipo que construye con el -g -O2 que es útil para depurar el software, y un "Release" tipo que construye con -O3, que es el predeterminado.


CONCLUSIONES

- ✓ PyBombs es una herramienta para ayudar a instalar GNU Radio, módulos fuera de árbol y otros paquetes de software, de origen o binarios, en su mayoría independientes del sistema.
- ✓ PyBombs permite el uso de prefijos personalizados le permite desinstalar versiones más fácilmente y tener varias versiones instaladas una junto a la otra.
- ✓ PyBOMBS proporciona mucho valor ya que es fácil de instalar y está estrechamente integrado en el proyecto y la comunidad de GNU Radio.

RECOMENDACIONES

- ✓ Verificar las especificaciones técnicas para lograr un óptimo funcionamiento.

PRÁCTICA 3.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Análisis de la tasa de error binario (BER)
<p>OBJETIVO GENERAL: Analizar la tasa de error binario de una señal determinada a través de la utilización de GNU Radio y USRP B205mini-i.</p> <p>OBJETIVOS ESPECIFICOS:</p> <ul style="list-style-type: none">➤ Configurar el equipo USRP B205mini-i, para la detección de una señal del entorno.➤ Ejecutar GNU Radio para la demostración de las diversas pérdidas de bits que pueden darse en la transmisión de una señal.➤ Demostrar el comportamiento de la señal mediante la interfaz gráfica de GNU Radio.		
INSTRUCCIONES		
<p>3. Introducción</p> <p>BER Es también conocido como tasa de error binario, el cual es la cantidad de bits recibidos incorrectamente, con respecto al número total de bits que son enviados en un intervalo de tiempo específico. Es diferente a otras formas de evaluación, ya que el BER, te permite evaluar un sistema de principio a fin, esto incluye el transmisor, el receptor y también el medio que existe entre ambos. De esta manera, el BER permite ejecutar el test para determinar el performance del sistema operando en la realidad, en lugar de elaborar el test de las partes por separado, con la perspectiva de que el comportamiento sea igual cuando estén en operación.</p> <p>Las causas que modifican el valor del BER son aquellos que hagan que la relación de señal ruido (S/N) en el trayecto de la RF pueda decaer, más las razones que puedan llegar afectar a las señales modulantes y las señales moduladas, serán factores que ampliarán el valor del BER. Tenemos, por ejemplo, la manifestación de señales que causen interferencia, el descenso de la potencia emitida por los transmisores, la sensibilidad de los receptores, insuficiencia de los equipos electrónicos,</p>		

en las antenas y en las guías de onda, el desenfoque de las parábolas, las nuevas edificaciones luego de la instalación de los enlaces que obstaculicen la trayectoria del haz de RF.

4. Actividades por desarrollar

Paso 1

Para crear el módulo gr-ofdm module se necesitan los siguientes pre requisitos

- ZeroMQ including the C++ and Python bindings (<http://zeromq.org/>)
- UHD Driver for USRP hardware
- GNU Radio (<http://gnuradio.org/>)
- ITPP (<http://itpp.sourceforge.net/>)
- python-qwt5-qt4
- python-scipy

Paso 2

Descargar de la siguiente dirección el repositorio <https://github.com/rwth-ti/gr-ofdm.git>

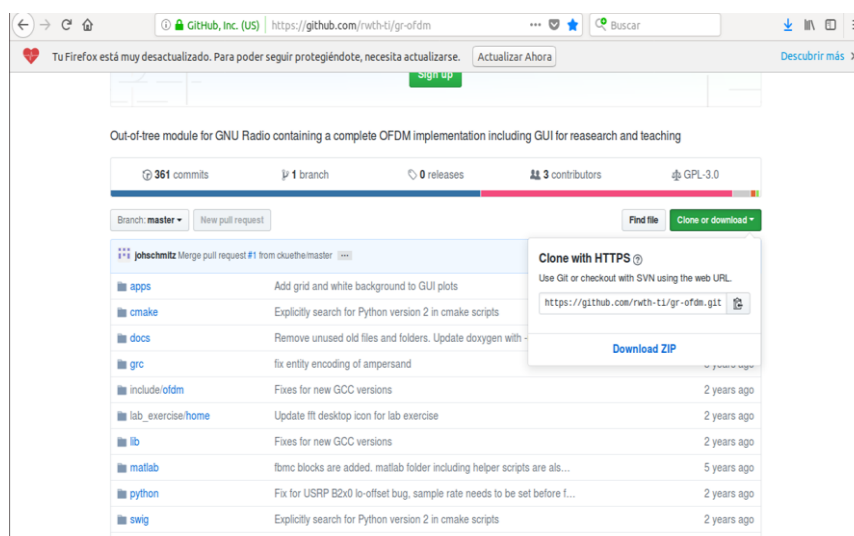


Figura 42 Descarga de Software

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Instalar la librería en la carpeta gr-ofdm con el siguiente código:

```
mkdir build
cd build
cmake ../
make
make install
```

Paso 4

Ejecutar los siguientes programas en el directorio de la carpeta gr-ofdm desde nuestro terminal:

```
/home/jaime/gr-ofdm/apps
```

```
run_usrp_tx_gui.sh
```

```
run_usrp_rx_gui.sh
```

```
jaime@jaime-SVE14A25CLH:~/Escritorio$ cd ..
```

```
jaime@jaime-SVE14A25CLH:~/ $ cd gr-ofdm
```

```
jaime@jaime-SVE14A25CLH:~/gr-ofdm$ cd apps
```

```
jaime@jaime-SVE14A25CLH:~/gr-ofdm/apps$ run_usrp_tx_gui.sh^C
```

```
jaime@jaime-SVE14A25CLH:~/gr-ofdm/apps$
```

Estos dos archivos nos permitirán ejecutar nuestra aplicación.

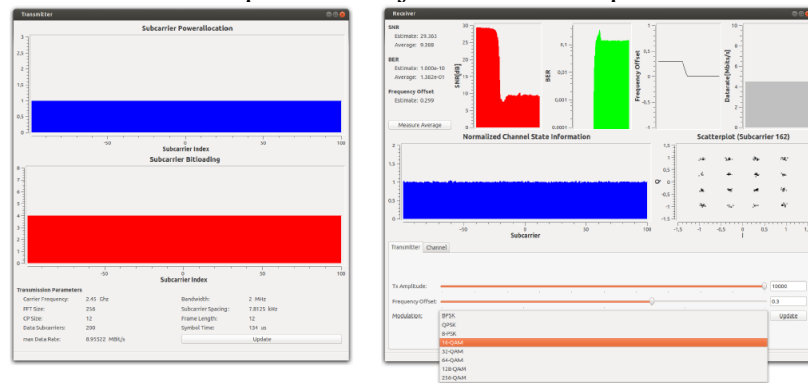


Figura 43 Archivos previas a la ejecución de la aplicación

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho


CONCLUSIONES

- ✓ Se observa una BER del sistema OFDM de dos maneras, con y sin la estimación de canal y, como consecuencia, se dibuja un gráfico, que muestra que la tasa de errores de bits del sistema OFDM con la estimación de canal es menor en comparación con uno sin estimación de canal.
- ✓ Se demostró el comportamiento de las señales, mediante el análisis de la tasa de error binario, mediante el entorno gráfico de GNU Radio.
- ✓ Se evidencia visualmente el cambio en la frecuencia y la ganancia de la señal mediante el entorno gráfico de GNU Radio.

RECOMENDACIONES

- ✓ Considerar los parámetros de potencia para evitar daños al equipo.
- ✓ Previo a la ejecución de la práctica, se debe tener instaladas las librerías instaladas.
- ✓ Se puede realizar investigación, utilizando diferentes esquemas de modulación, como QPSK, QAM, 8QAM, etc. en lugar de BPSK y elegir los mejores esquemas de sincronización y estimación de canal.

PRÁCTICA 4.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Codificación de canales
OBJETIVO GENERAL: Analizar la frecuencia de las diferentes codificaciones de los canales que se utilizan en una transmisión de señales. OBJETIVOS ESPECIFICOS: ➤ Registrar los parámetros de configuración del equipo URSP. ➤ Ejecutar GNU Radio para la observación gráfica de la frecuencia de señal de un canal determinado.		
INSTRUCCIONES		
1. Introducción Codificación de canales OFDM En todo sistema de comunicación hay un objetivo en común: la transmisión de la información que hay desde el emisor hasta llegar al receptor por medio de un canal. Debido a que existe ruido en todo canal, el mensaje enviado presenta una distorsión en el trayecto hacia el receptor, por lo que en la mayoría de los casos se producen diferencias en las secuencias de datos que son enviados y los datos recibidos, estas diferencias se las denomina errores. Para sustentar estos errores en la recepción de datos, se realiza la codificación de canal, cuyo fin es que el receptor detecte para así corregir los errores que se producen durante la transmisión de datos. La codificación de canal se basa en incluir redundancia, de forma que sea factible la reconstrucción de la secuencia de datos originales de la forma más confiable posible. El sistema de comunicación OFDM consiste en un modelo de canal a través del cual los símbolos de datos se transmiten al receptor. Este modelo de		

canal produce comunicación de línea de visión y también varias reflexiones debido a que el efecto de trayectos múltiples aparece en la imagen. Para minimizar el efecto multitrayecto y el ruido introducido por el canal, optamos por la estimación del canal. En general, la señal recibida obtenida a través de un canal de desvanecimiento multitrayecto selectivo de frecuencia contiene la respuesta de impulso del canal y el ruido Gaussiano blanco aditivo.

2. Actividades por desarrollar

Paso 1

Abrir el terminal y escribir `gnuradio-companion`

Se abrirá GNU RADIO

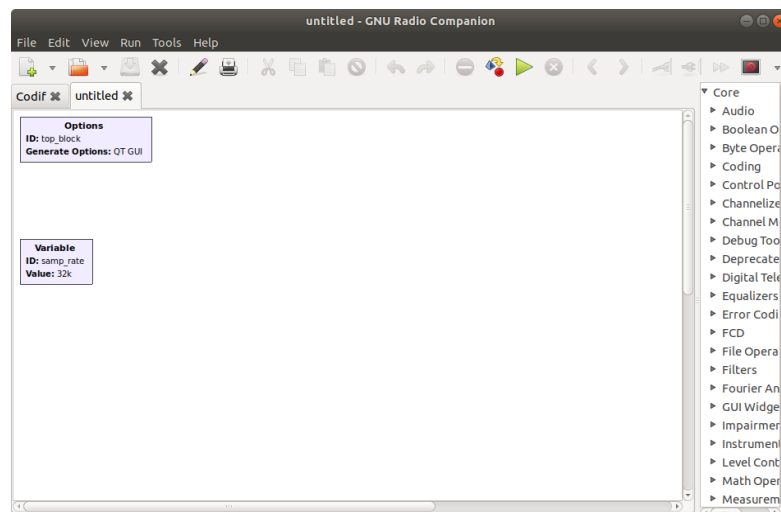
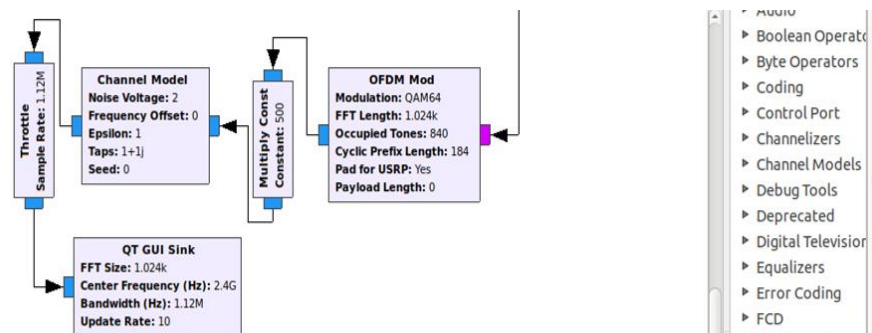


Figura 44 GNU Radio

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 2: Colocamos los siguientes bloques



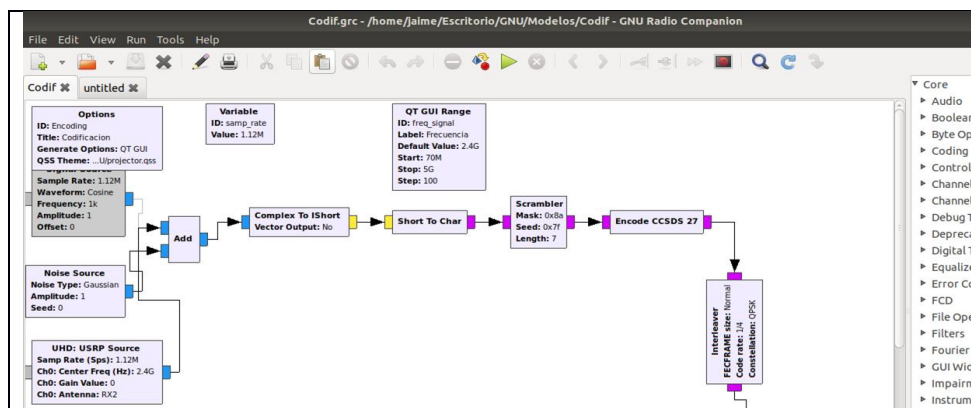


Figura 45 GNU Radio: bloques

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Figura 46 Bloque Propiedades

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque variables

Este bloque mapea un valor para una única variable. Este bloque variable no tiene representación gráfica.

Pondremos los siguientes parámetros:

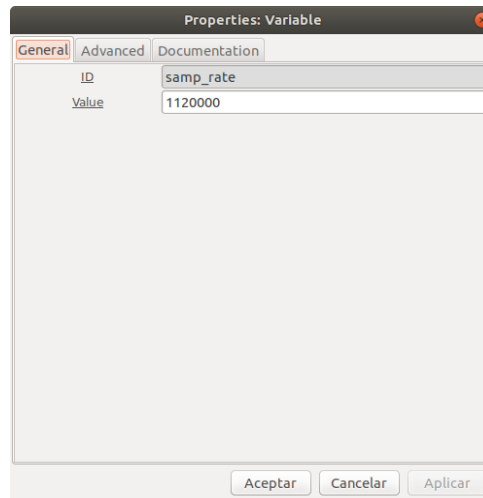


Figura 47 Bloque Variables

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque QT GUI RANGE

Este bloque crea una variable con una slider. Deja la etiqueta en blanco para usar la variable id como la etiqueta. El valor tiene que ser un número real. El valor tiene que estar entre el inicio y la pausa.

Se puede utilizar una sugerencia para posicionar el widget dentro de la aplicación. La sugerencia es de la forma [tab_id @ tab_index]: [row, col, row_span, col_span]. Tanto la especificación de la pestaña como la posición de la cuadrícula son opcionales.

Bloque Signal Source

Este bloque realiza funciones de generador de señales

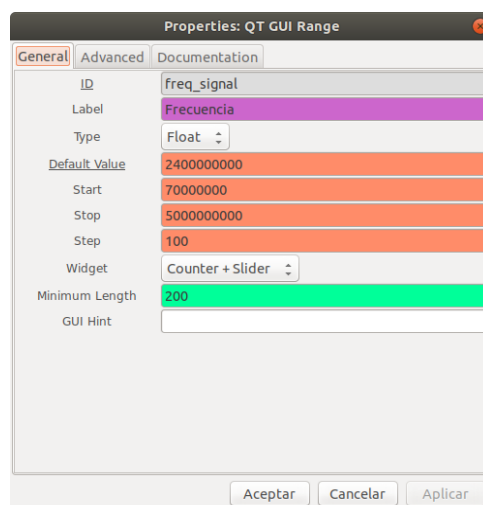


Figura 48 Bloque Signal Source

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

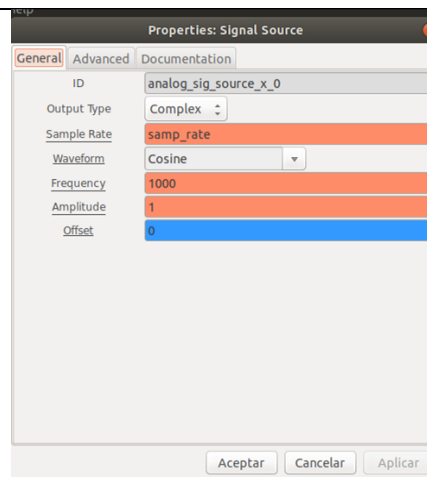


Figura 49 Funciones de generador de señales
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Noise source

Genera valores aleatorios a partir de diferentes distribuciones. Actualmente, solo gaussianos y uniformes están habilitados.

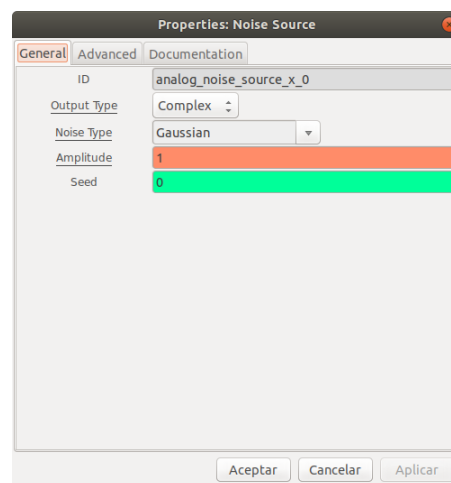


Figura 50 Bloque Noise source
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Add

Agrega muestras en todos los flujos de entrada. Para todas las muestras en todos los flujos de entrada.

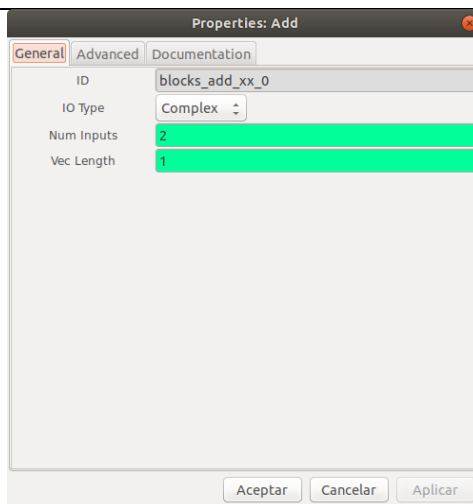


Figura 51 Bloque Add

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque UHD USRP Source

El bloque de fuente UHD USRP:

Dirección del dispositivo:

La dirección del dispositivo es una cadena delimitada que se utiliza para localizar dispositivos UHD en su sistema. Si se deja vacío, será usado el primer dispositivo UHD encontrado. Utilice la dirección del dispositivo para especificar un dispositivo específico o una lista de dispositivos.

Ejemplo de USRP1: serial = 12345678

Ejemplo de USRP2: addr = 192.168.10.2

Ejemplo de USRP2: addr0 = 192.168.10.2, addr1 = 192.168.10.3

Out type:

Este parámetro controla el tipo de datos del flujo en gnuradio.

Wire format:

Este parámetro controla la forma de los datos a través del bus / red. Los bytes complejos se pueden usar para intercambiar precisión por ancho de banda. No todos los formatos son compatibles con todos los dispositivos.

Stream Args:

Argumentos opcionales que se pasarán en el objeto streamer UHD. Streamer args es una lista de pares clave / valor; El uso está determinado

por la implementación.

Ejemplo: la clave escalar afecta la escala entre enteros de 16 y 8 bits en formato de cable sc8.

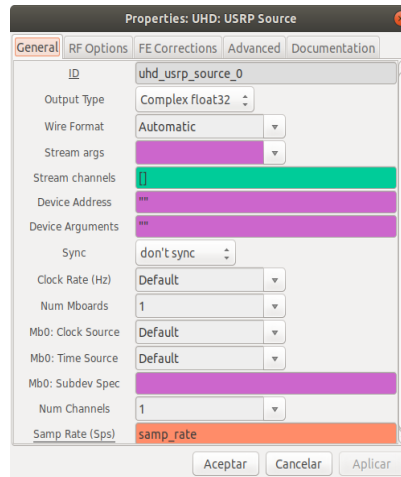


Figura 52 Propiedades UHD: USRP Source

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

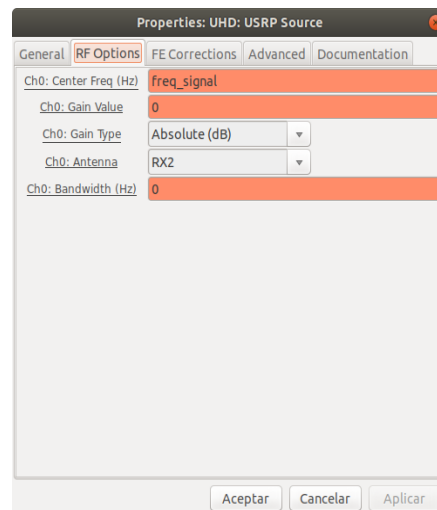


Figura 53 Propiedades UHD

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Throttle

Flujo de aceleración de muestras de manera que la tasa promedio no exceda a samples_per_sec.

entrada: una secuencia de itemsize; salida: una secuencia de itemsize

nótese bien esto solo debe usarse en aplicaciones GUI donde no hay otro bloque limitador de velocidad. No tiene la intención ni es eficaz para controlar con precisión la tasa de muestras. Eso debe ser controlado por una fuente o sumidero vinculado al reloj de muestra. Por ejemplo, una

tarjeta de audio o USRP.

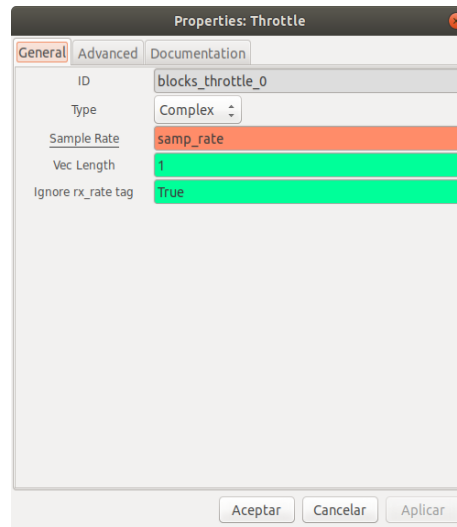


Figura 54 Bloque Throttle

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Channel Mode

Simulador de canal básico.

Este bloque implementa un simulador de modelo de canal básico que puede usarse para ayudar a evaluar, diseñar y probar varias señales, formas de onda y algoritmos.

Este modelo permite al usuario configurar el voltaje de una fuente de ruido AWGN (), un desplazamiento de frecuencia (normalizado), un desplazamiento de tiempo de muestra () y una semilla () para aleatorizar o hacer reproducible la fuente de ruido AWGN. Las rutas múltiples se pueden aproximar en este modelo utilizando una representación de filtro FIR de un perfil de retardo de múltiples rutas con el parámetro.

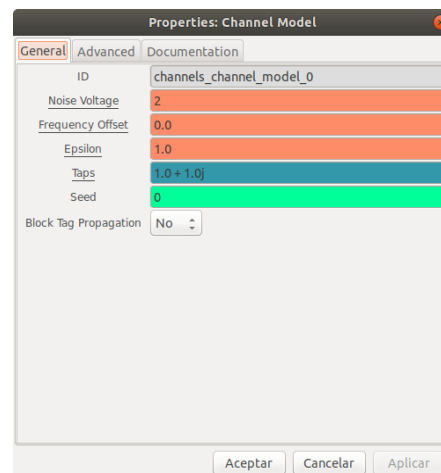


Figura 55 Bloque Channel Mode

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Multiply Const

Crea una instancia del multiply_const_vcc.

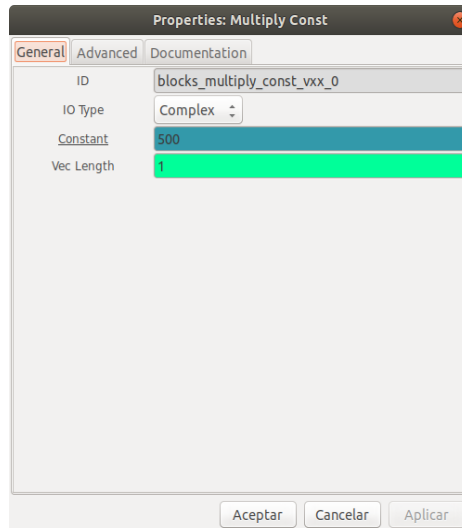


Figura 56 Bloque Multiply Const

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Mod

Modula un flujo OFDM. Basado en las opciones `fft_length`, `occup_tones`, y `cp_length`, este bloque crea símbolos OFDM usando una opción de modulación específica.

Envía paquetes llamando a `send_pkt`

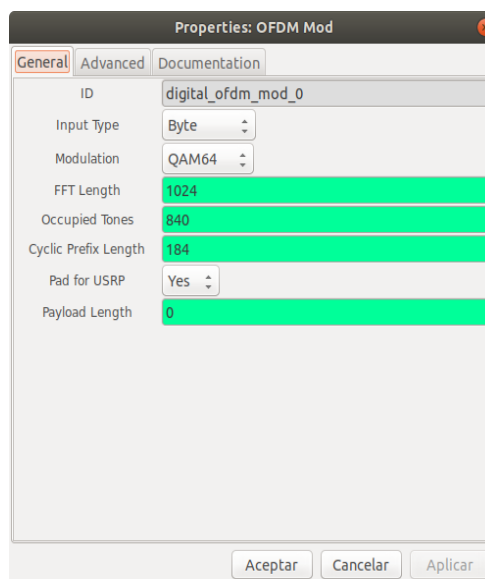


Figura 57 Bloque OFDM Mod

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque QT GUI Sink

GUI Hint se puede utilizar para colocar el widget dentro de la aplicación. La sugerencia es de la forma [tab_id @ tab_index]: [row, col, row_span, col_span]. Tanto la especificación de la pestaña como la posición de la cuadrícula son opcionales.

Este es un sumidero gráfico basado en QT que toma un flujo flotante y lo grafica. La acción predeterminada es trazar la señal como PSD (FFT), espectrograma (cascada) y gráficos en el dominio del tiempo. Las parcelas pueden desactivarse estableciendo el valor booleano apropiado en el constructor en Falso.

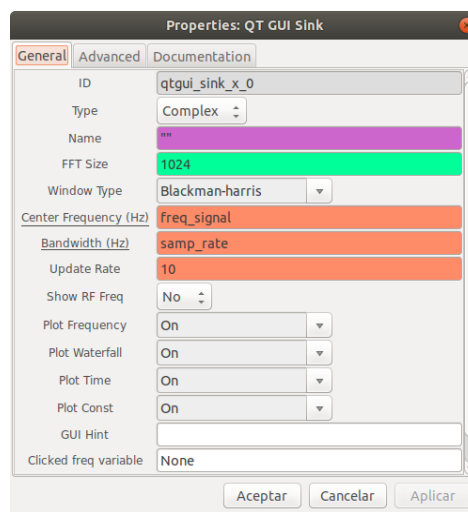


Figura 58 Bloque QT GUI Sink

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Interleaver

Bloque intercalador

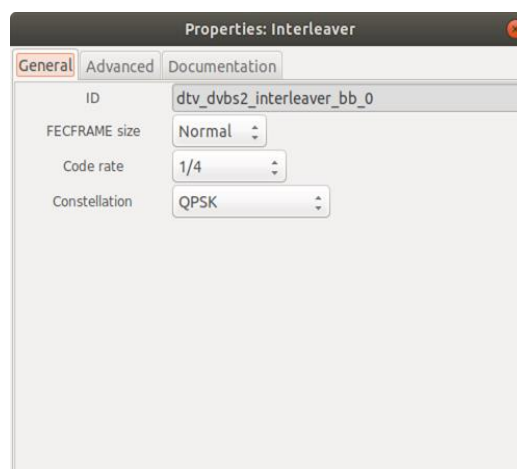


Figura 59 Bloque Interleaver

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Ejecutamos la aplicación

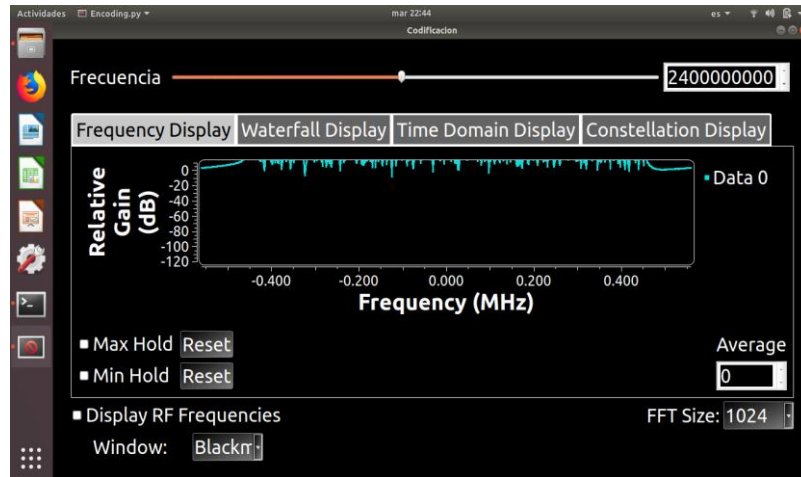


Figura 60 Ejecución de aplicación

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho


CONCLUSIONES

- ✓ La ventaja de utilizar radio definido por software es que se puede diseñar y probar diferentes configuraciones en el menor tiempo posible.
- ✓ Se pueden desarrollar algunos bloques de procesamiento para mejorar la detección de símbolo en el receptor, los bloques con los que se trabajaron se configuraron a la necesidad del sistema emulado, sin embargo, esto abre una oportunidad para el desarrollo de propios bloques o la creación de una completa biblioteca para la simulación y emulación de sistemas de telecomunicaciones.

RECOMENDACIONES

- ✓ Incrementar las capacidades del equipo de cómputo a utilizar para la emulación, de un bus USB de 8MHz (2.0) a 32MHz (3.0) para utilizar un mayor ancho de banda en la transmisión.
- ✓ Mejorar las etapas de codificación y estimación de canal (completa implementación).

PRÁCTICA 5.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	5	TÍTULO PRÁCTICA: Radar
OBJETIVO GENERAL: Analizar el espectro de una señal de radar para determinar la distancia de objetos que reflejan dicha señal. OBJETIVOS ESPECIFICOS: ➤ Proporcionar un entorno genérico en GNU Radio para experimentar con varios tipos de radar.		
INSTRUCCIONES		
1. Introducción El flujo de trabajo para una configuración de radar sincronizada: primero conecta todo el hardware y luego envíe una señal de pulso que debe ser reflejada por un objetivo cercano, por ejemplo, una pared. Un generador de señal para los pulsos de sincronización está en la caja de herramientas incluida. El estimador correlaciona de forma cruzada estas dos señales y busca la mejor coincidencia con respecto a un cambio de señal específico. Este número de muestras desplazadas se proporciona en una salida de consola mediante el bloque de resultados de impresión o a través de la interfaz gráfica de usuario QT incorporada. 2. Actividades por desarrollar Paso1 Debe tener instaladas las siguientes dependencias Qt 4.8.6 Qwt 6.0.0		

python-matplotlib

Qt y Qwt deben ser instaladas con GNU Radio. Pero puede haber problemas si Qwt está por debajo de la versión 6. python-matplotlib se incluye en algunos casos de prueba.

Para instalar Qt 4.8.6

```
1 $ sudo apt-get install -y build-essential
2 $ sudo apt-get build-dep qt4-qmake
3
4 $ wget http://download.qt.io/archive/qt/4.8/4.8.6/qt-everywhere-opensource-src-
5 $ tar -xzf qt-everywhere-opensource-src-4.8.6.tar.gz
6 $ cd qt-everywhere-opensource-src-4.8.6
7
8 # En el configure nos pedirá elegir entre la version open source o privada,
9 # yo elegi la open source.
10 $ ./configure
11
12 # Esto llevará un buen rato
13 $ make
14
15 $ sudo make install
```

```
1 # Guardo una copia de qmake
2
3 $ sudo mv /usr/bin/qmake /usr/bin/qmake.old
4 $ sudo /usr/local/Trolltech/Qt-4.8.6 /usr/bin/qmake
```

Figura 61 Instalación de Qt 4.8.6

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Para instalar Qwt 6.0.0

Ir a la siguiente dirección:

https://osdn.net/projects/sfnet_qwt/downloads/qwt/6.0.0/qwt-6.0.0.zip/

Descargar qwt-6.0.0.zip ([qwt-6.0.0.zip \(external link: SF.net\)](#): 3,687,942 bytes).

Para instalar python-matplotlib

Ejecutar en el terminal lo siguiente:

```
sudo apt-get install python3-matplotlib
```

Ingresa los siguientes comandos en su terminal.

```
git clone https://github.com/kit-cel/gr-radar.git// clonar este repositorio
```

```
cd gr-radar/
```

```
mkdir build// crear una carpeta de
```

```
cd build/
```

```
cmake ../compilación // crear makefiles
```

```
make// crear una caja de herramientas
```

Algunos casos de prueba utilizan tamaños de tampones enormes. Por lo tanto, debe aumentar el tamaño máximo de un único segmento de memoria compartida. Ejecute la siguiente línea para arreglar eso. Además, el script configura algunas cosas USRP. Todos los cambios se restablecerán después de reiniciar.

```
sudo ../examples/setup/setup_core
```

Ahora puedes ejecutar todos los testcases e instalar la caja de herramientas. Si corresponde con 100% de pruebas aprobadas', la caja de herramientas se debe construir correctamente.

```
ctest// ejecutar pruebas
```

```
sudo make install// instalar caja de herramientas
```

Paso 2

Ejecutamos la aplicación

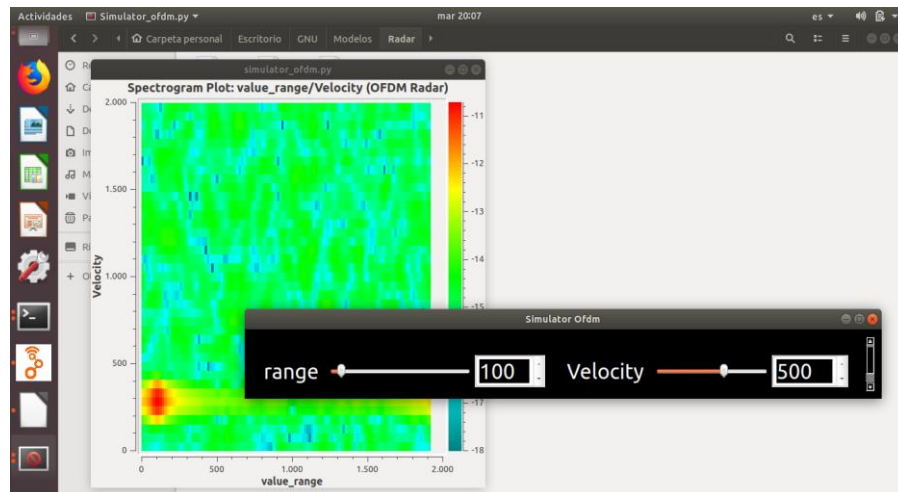


Figura 62 Ejecución de aplicación

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

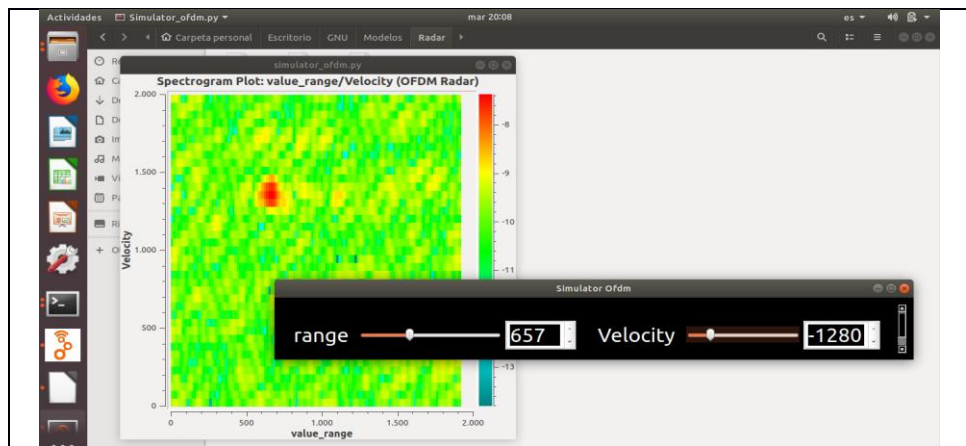


Figura 63 Calibración de parámetros

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho


CONCLUSIONES

- ✓ El receptor GND Radio OFDM puede servir como base para una mayor experimentación, mediciones e investigación en algoritmos de procesamiento de señales.
- ✓ Permite reproducibilidad de estudios conceptuales y experimentos, y todos los bloques del receptor se pueden analizar con más detalle mediante compañeros investigadores

RECOMENDACIONES

- ✓ Cuando se investigan técnicas de SDR por primera vez, la radio GNU Se recomienda instalar el software y el controlador de hardware USRP en el Software VMware (o cualquier otro software similar) para configuraciones simples y modificaciones;
- ✓ Las computadoras más potentes con tarjetas Gigabit Ethernet deben usarse para mejorar el rendimiento del sistema de comunicación implementado;
- ✓ Se podría utilizar una frecuencia de referencia externa más precisa, como el GPS, para mejorar la precisión de la sincronización;
- ✓ Podrían utilizarse atenuadores de potencia para resolver los problemas tempranos de saturación de potencia en los receptores, por lo que se podría ampliar el rango experimental de SNR.

PRÁCTICA 6.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Sincronización de canales
OBJETIVO GENERAL: Analizar la sincronización de canales a través de GNU Radio y USRP B205mini-i. OBJETIVOS ESPECIFICOS: ➤ Ingreso de parámetros del canal de transmisión para la recepción de una señal. ➤ Analizar la ganancia y frecuencia del canal utilizado.		
INSTRUCCIONES		
1. Introducción Sincronización de canales OFDM La sincronización se basa en la búsqueda de un símbolo de entrenamiento con dos mitades idénticas en el dominio del tiempo, que permanecerán igual después de pasar a través del canal, excepto que habrá una diferencia de fase entre ellas causada por el desplazamiento de frecuencia de la portadora. Para el caso de GNU-Radio se utilizan 2 símbolos o palabras. En nomenclatura de OFDM, un símbolo se refiere a todas las sub portadoras (FFT). El primer símbolo está compuesto por dos mitades y se construyen igual (en orden de tiempo); mediante la transmisión de una secuencia de pseudoruido (PN) en las frecuencias pares, mientras que los ceros se usan en las frecuencias impares. Dicho de otra forma, en cada frecuencia par, uno de los puntos de una constelación QPSK es transmitido. Con el fin de mantener la energía de la señal aproximadamente constante para cada símbolo de los componentes de la frecuencia de este símbolo de entrenamiento, se multiplica.		

En todo sistema de comunicación la sincronización juega un papel fundamental, pues gracias a ella seremos capaces de recuperar correctamente la información. Una falta de sincronización, tanto en tiempo como en frecuencia o en fase, daría lugar a una serie de errores en la demodulación de la señal recibida. Para empezar, una pequeña desviación de nuestra frecuencia provoca la pérdida de la ortogonalidad de la señal OFDM, lo que lleva a un muestreo en frecuencia incorrecto dando lugar a la Interferencia entre portadoras (ICI). Por otra parte, las desviaciones en fase provocan una rotación de los símbolos de la constelación, haciendo que el decisor realice la asociación símbolo-bits de manera errónea. Siguiendo las indicaciones del estándar IEEE 802.11a, cada paquete contiene unos preámbulos PLCP (Protocolo de convergencia de capa física), mostrados en la figura 1.7, cuya función, entre otras muchas, es la de sincronización.

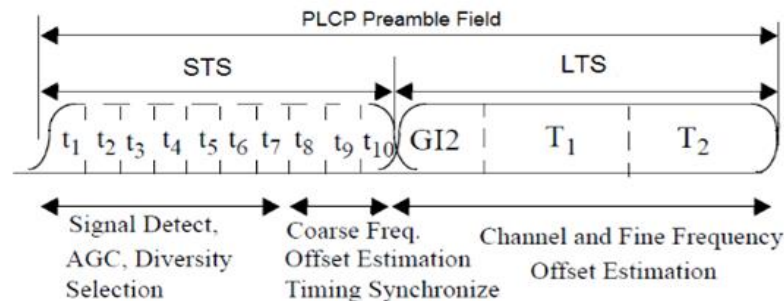


Figura 64 Campo de preámbulo

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

2. Actividades por desarrollar

Paso 1

Instalar lo siguiente desde el terminal

```
git clone git@github.com:gnuradio/gr-inspector.git
```

```
cd gr-inspector
```

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make -j4
```

```
sudo make install
```

Paso 2

Abrir el terminal y escribir gnuradio-companion

Se abrirá GNU RADIO

Paso 3

Crear el grc

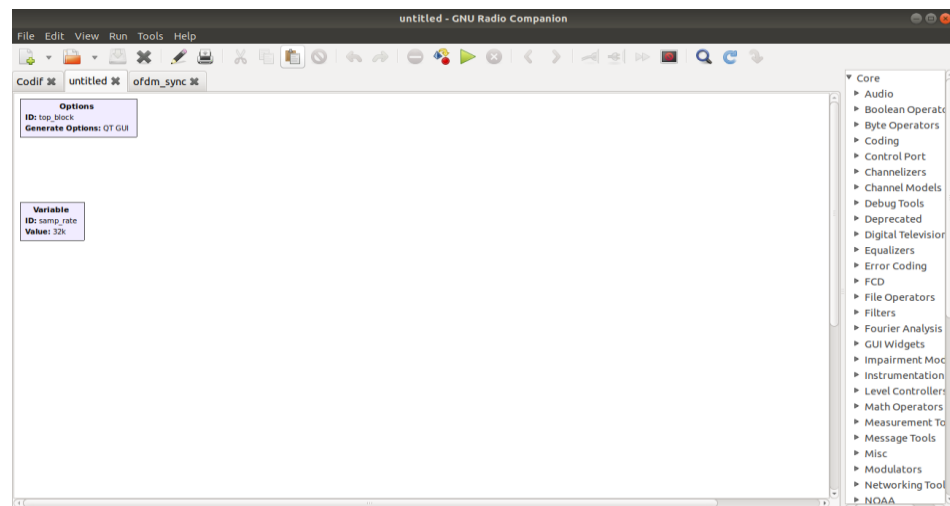


Figura 65 Creación de GRC

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

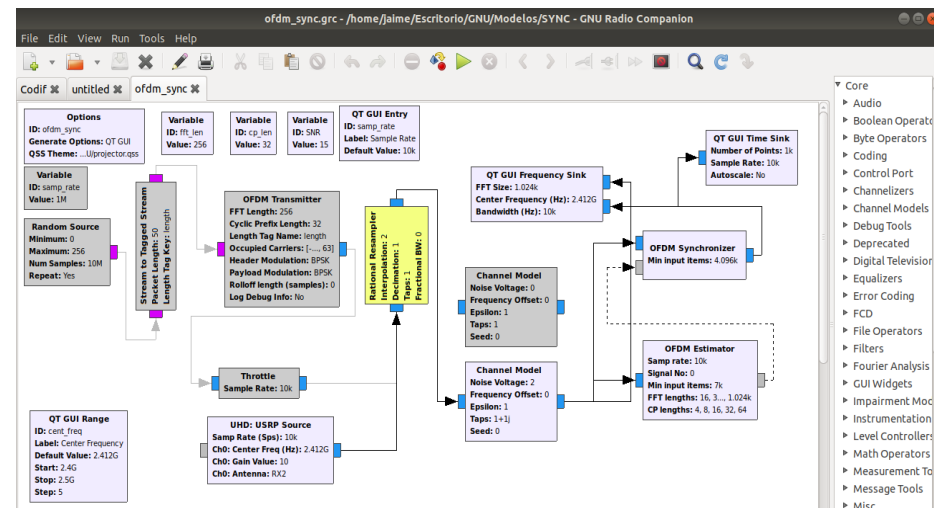


Figura 66 Proceso de creación de GRC

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

QT GUI Range
ID: cent_freq
Label: Center Frequency
Default Value: 2.412G
Start: 2.4G
Stop: 2.5G
Step: 5

QT GUI Range
ID: gain
Label: Ganancia
Default Value: 10
Start: 0
Stop: 100
Step: 1

Figura 67 Rangos QT GUI

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetros del bloque opciones

The image shows a 'Properties: Options' dialog box with three tabs: 'General', 'Advanced', and 'Documentation'. The 'General' tab is active. It contains the following fields and controls:

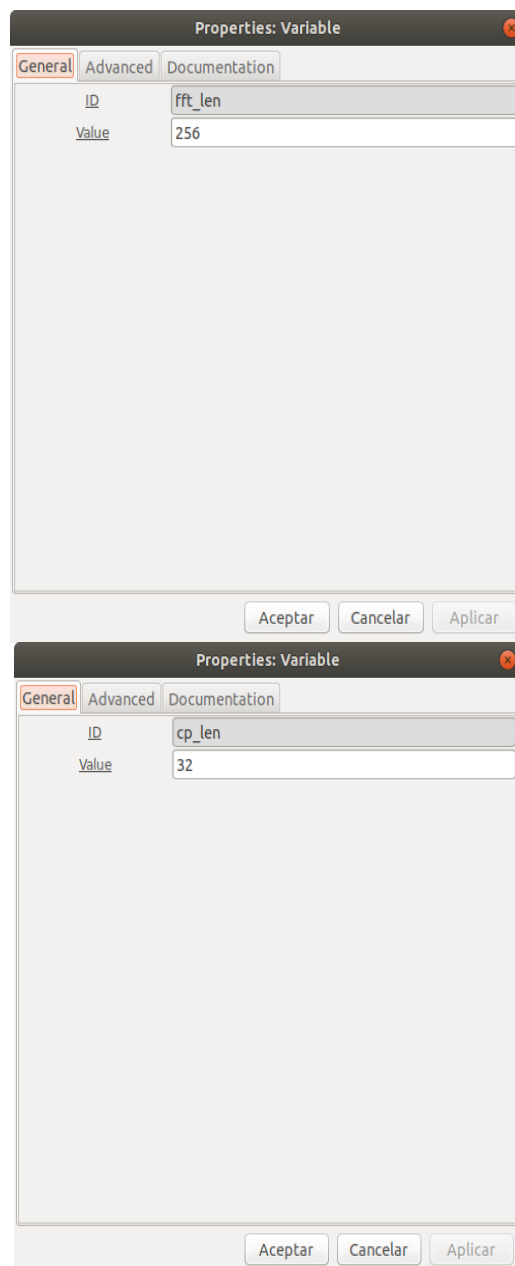
- ID:** ofdm_sync
- Title:** (empty text field)
- Author:** (empty text field)
- Copyright:** (empty text field)
- Description:** (empty text field)
- Canvas Size:** (empty text field)
- Generate Options:** QT GUI (dropdown menu)
- Run:** Autostart (dropdown menu)
- Max Number of Output:** 0
- Realtime Scheduling:** Off (dropdown menu)
- QSS Theme:** /home/jaime/Escritorio/GNU/projector.qss (text field with a browse button '...')

At the bottom of the dialog are three buttons: 'Aceptar', 'Cancelar', and 'Aplicar'.

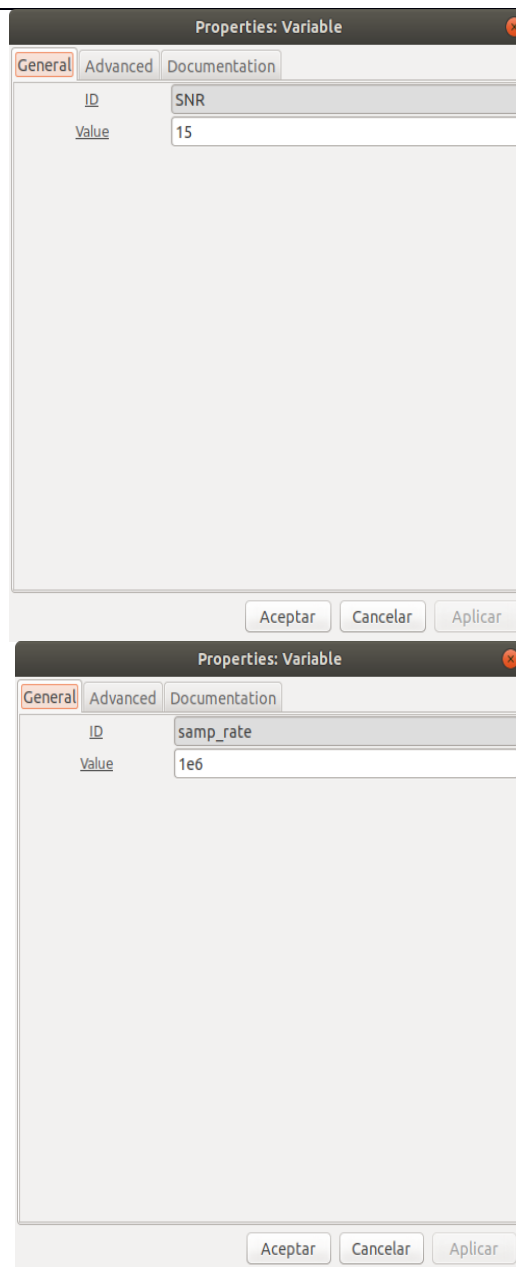
Figura 68 Parámetros del bloque opciones

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Variables que crearemos



Figuras 69 y 70 Creación de variables `fft_len` y `cp_len` respectivamente
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho



Figuras 71 y 72 Variables SNR y samp_rate respectivamente
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque QT GUI Entry

Este bloque crea una variable con una caja de entrada de texto. Deja la etiqueta en blanco para usar la variable id como la etiqueta.

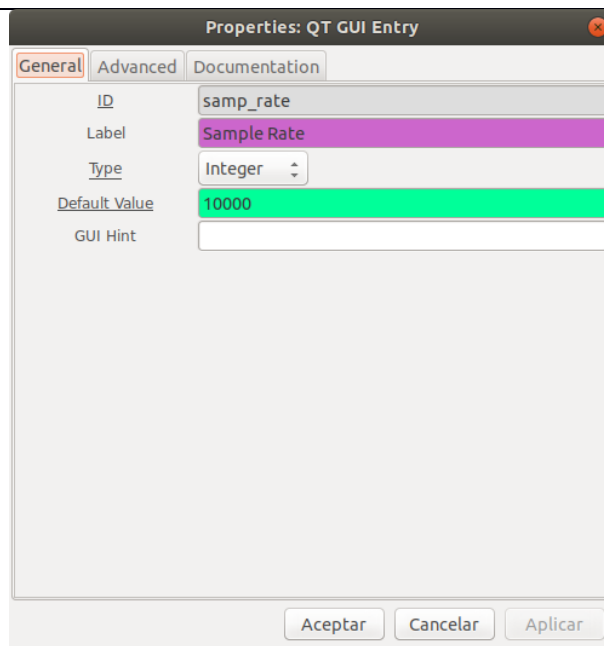


Figura 73 Bloque QT GUI Entry

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Random Source

Genera num samples de números aleatorios de [min, max). Repite las muestras si se especifica.

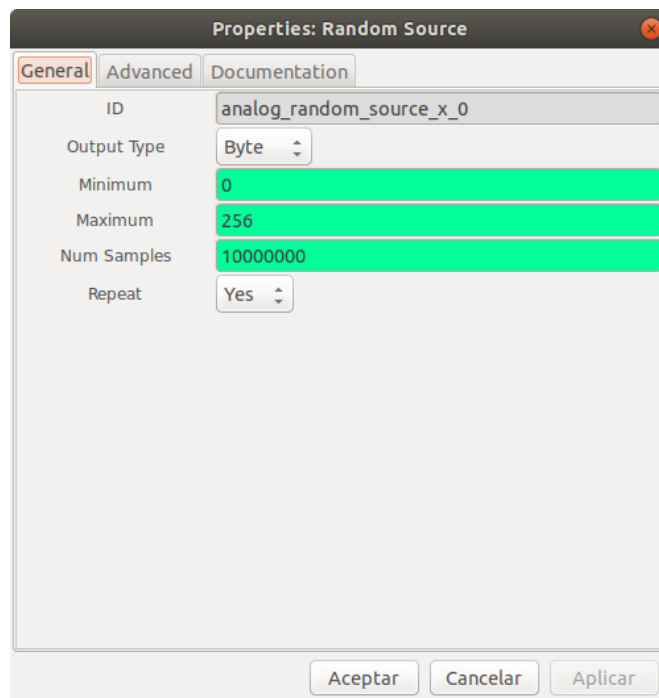


Figura 74 Bloque Random Source

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Stream to Tagged Stream

Convierte una secuencia normal en una secuencia etiquetada.

Todo lo que hace este bloque es agregar etiquetas de longitud en intervalos regulares. Se puede usar para conectar una transmisión regular a `gr :: tagged_stream_block`.

Este bloque está destinado a conectarse directamente a un bloque de flujo etiquetado. Si hay bloques entre este bloque y un bloque de flujo etiquetado, asegúrese de que no cambien la velocidad, o modifique el valor de la etiqueta para asegurarse de que las etiquetas de longitud representan la longitud del paquete.

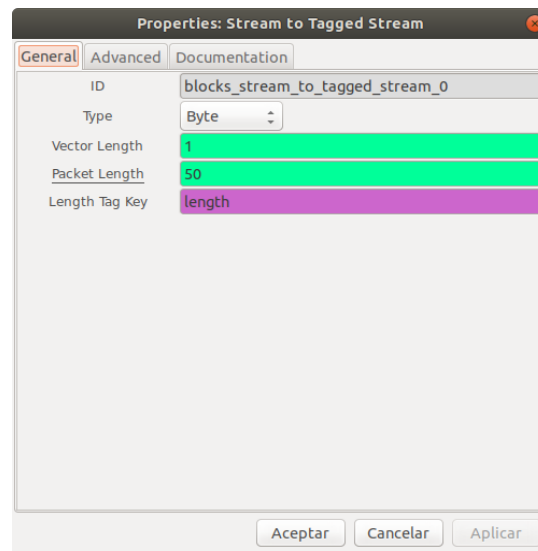


Figura 75 Bloque Stream to Tagged Stream

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Transmitter

Bloque jerárquico para la modulación OFDM.

La entrada es un flujo de bytes (caracteres sin signo) y la salida es la señal modulada compleja en banda base.

Property	Value
ID	digital_ofdm_tx_0
FFT Length	fft_len
Cyclic Prefix Length	cp_len
Length Tag Name	length
Occupied Carriers	(range(-128/2,0)+range(1,128/2),)
Pilot Carriers	()
Pilot Symbols	()
Sync Word 1	()
Sync Word 2	()
Header Modulation	BPSK
Payload Modulation	BPSK
Rolloff length (samples)	0
Scramble Bits	No
Log Debug Info	No

Figura 76 Bloque jerárquico para la modulación OFDM.
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Rational Resampler

Property	Value
ID	rational_resampler_xxx_0
Type	Complex->Complex (Complex Taps)
Interpolation	2
Decimation	1
Taps	1
Fractional BW	0

Figura 77 Bloque Rational Resampler
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

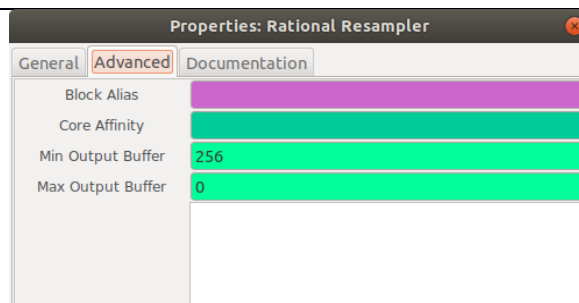


Figura 78 Propiedades de bloque Rational Resampler
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Synchronizer

Sincronizador de frecuencia OFDM y estimador de temporización.

Este bloque se puede insertar hacia abajo en el bloque de estimación de parámetros OFDM. Realiza una estimación de desplazamiento de frecuencia y una estimación de temporización de símbolos según [1]. Las correcciones de frecuencia más pequeñas que el espaciado de la subportadora se corregirán para obtener la ortogonalidad de las portadoras. La salida es un flujo con etiquetas de flujo en los inicios estimados del símbolo OFDM (símbolo "símbolo").

Conecte la salida del estimador de parámetros OFDM al puerto de mensajes y señálese a sí misma a la entrada de flujo.

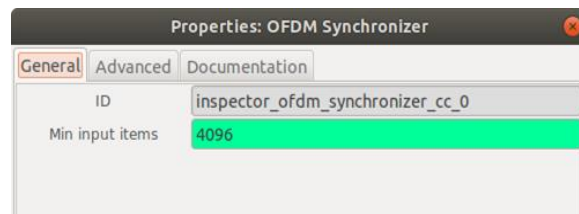


Figura 79 Bloque OFDM Synchronizer
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Estimator

Parámetro bloque de estimación para señales OFDM.

Este bloque realiza una estimación de parámetros de una señal OFDM de entrada. Los parámetros que se estiman son la longitud de FFT, la longitud del prefijo cíclico, el tiempo del símbolo y el espaciado de la subportadora. La entrada es un flujo complejo, la salida es un mensaje como tupel de tupels que contienen la descripción de los resultados del análisis y los valores.

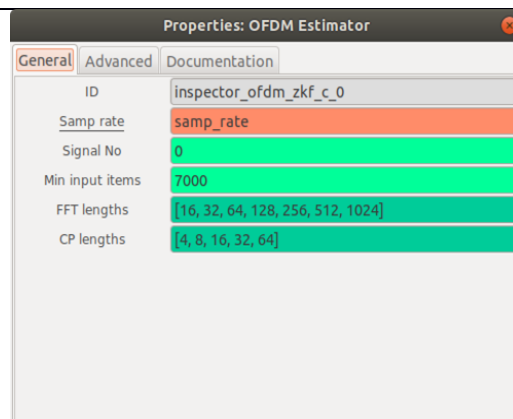


Figura 80 Bloque de estimación para señales OFDM
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetros del bloque UHD: USRP Source

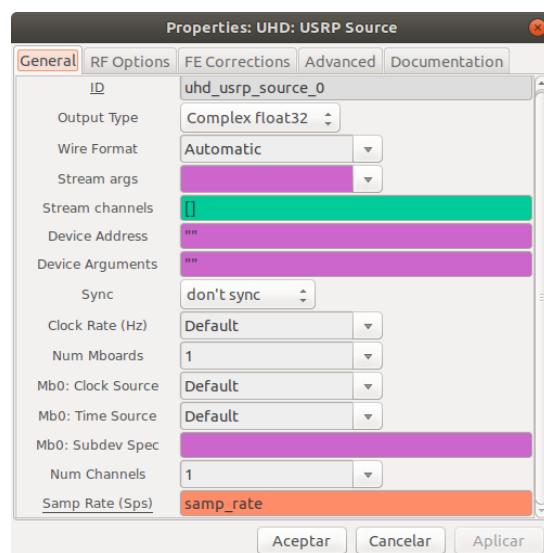


Figura 81 Parámetros del bloque UHD: USRP Source
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

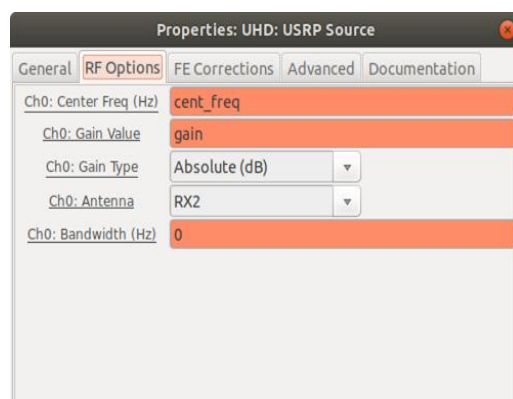
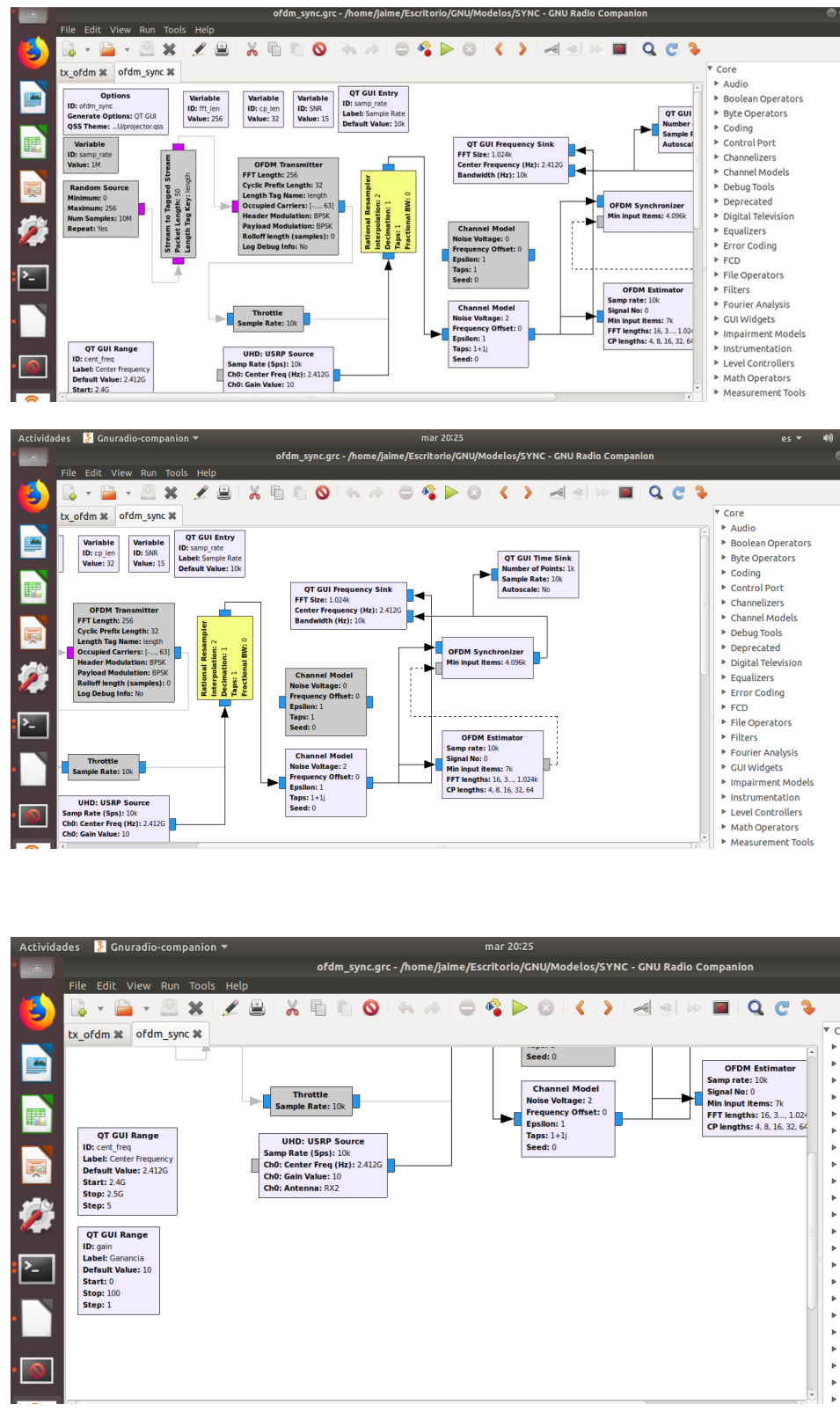
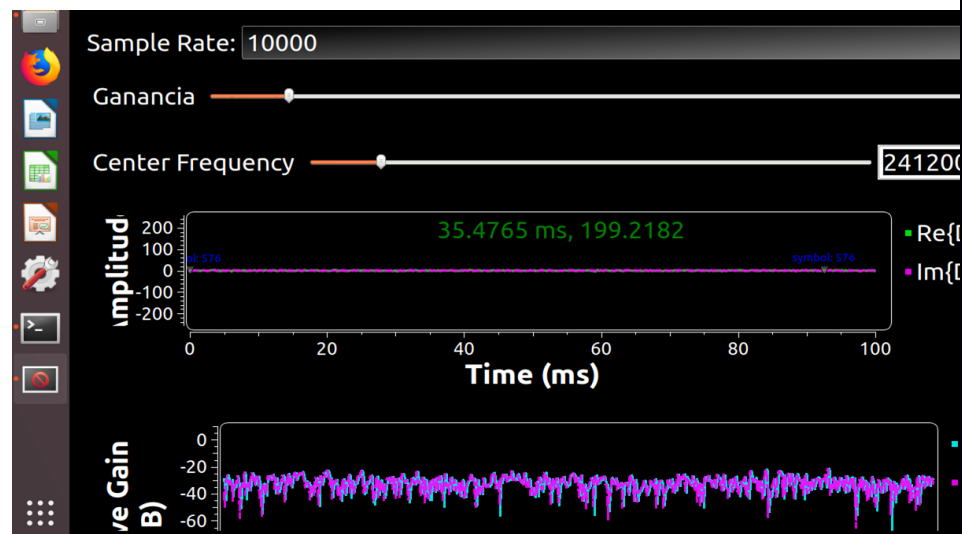


Figura 82 Propiedades del bloque UHD: USRP Source
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Ejecutamos nuestra aplicación





Figuras 83, 84, 85 y 86 Aplicación en ejecución
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho


CONCLUSIONES

- ✓ La técnica OFDM se usa ampliamente en aplicaciones en tiempo real como los sistemas de evaluación a largo plazo (LTE), WiFi, WiMax.
- ✓ La implementación de comunicaciones inalámbricas utilizando software definido por radio (SDR) es de código abierto y una de las áreas de investigación emergentes. SDR es una técnica para aplicaciones inalámbricas de alta velocidad con dispositivos de hardware mínimos y también rentables.
- ✓ En este documento, hemos implementado un sistema de comunicación basado en OFDM con modelo de canal, estimación de canal y sincronización utilizando GNU Radio, que es una fuente abierta.

RECOMENDACIONES

- ✓ Incrementar las capacidades del equipo de cómputo a utilizar para la emulación, de un bus USB de 8MHz (2.0) a 32MHz (3.0) para utilizar un mayor ancho de banda en la transmisión.
- ✓ Mejorar las etapas de codificación y estimación de canal (completa implementación).

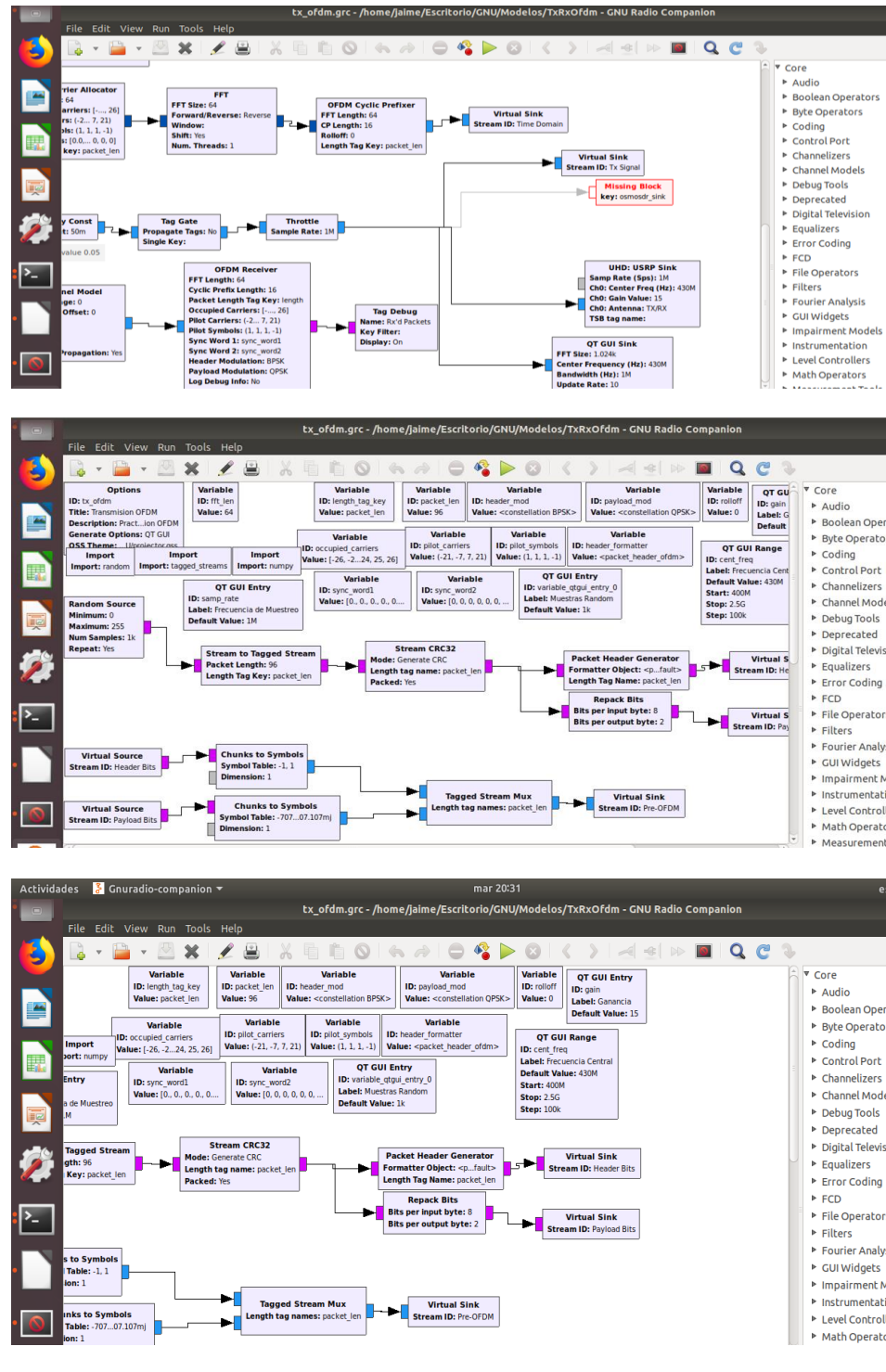
PRÁCTICA 7.

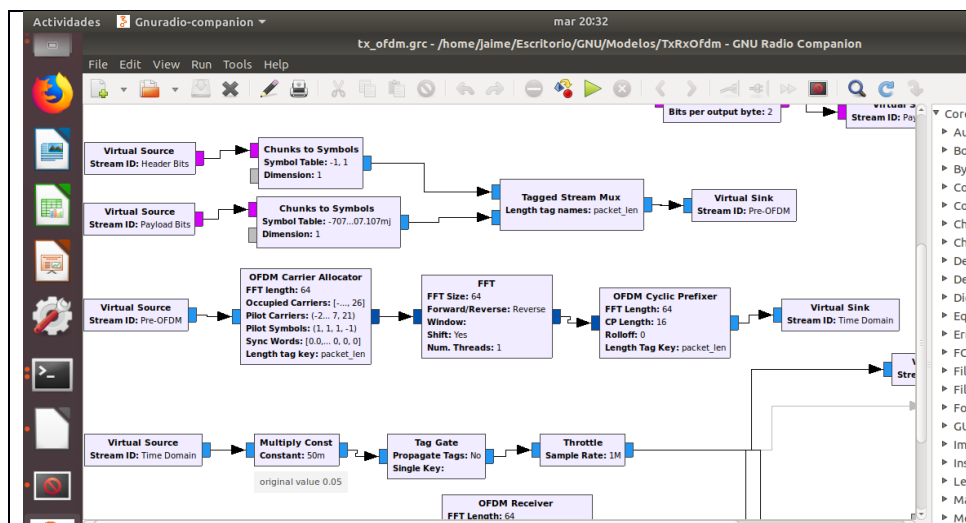
 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	7	TÍTULO PRÁCTICA: Transmisión y recepción OFDM
OBJETIVO GENERAL: Desarrollar GRC-flowgraphs para transmitir y recibir paquetes de datos a través de ortogonales. Multiplexación por división de frecuencia (OFDM) OBJETIVOS ESPECIFICOS: ➤ Construir bloques en GNU-Radio para el prefijo y la potencia cíclicos. ➤ Asignar en bandas de subportadoras.		
INSTRUCCIONES		
Paso 1 Instalar lo siguiente desde nuestro terminal: <code>git clone https://github.com/tfcollins/gr-ofdm.git</code> <code>cd gr-ofdm</code> <code>mkdir build</code> <code>cd build</code> <code>cmake ..</code> <code>make -j4</code> <code>sudo make install</code>		

Paso 2

Crear el grc

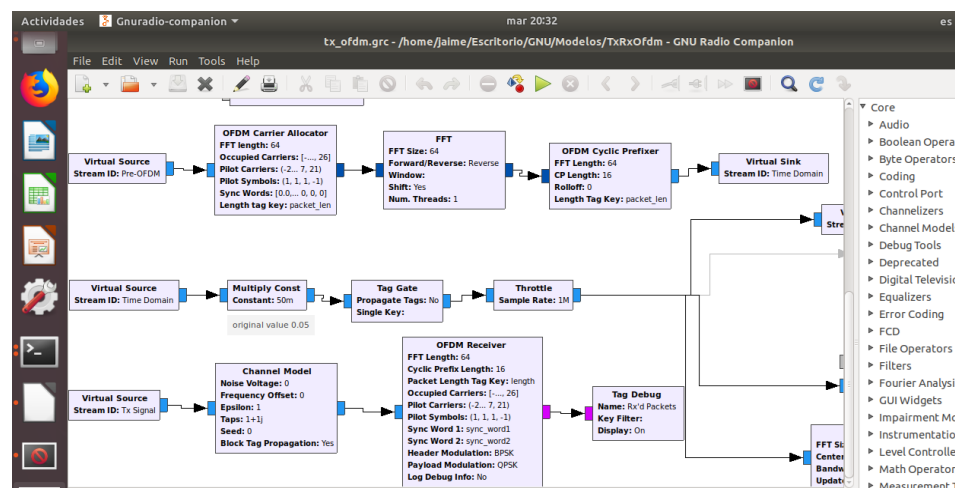
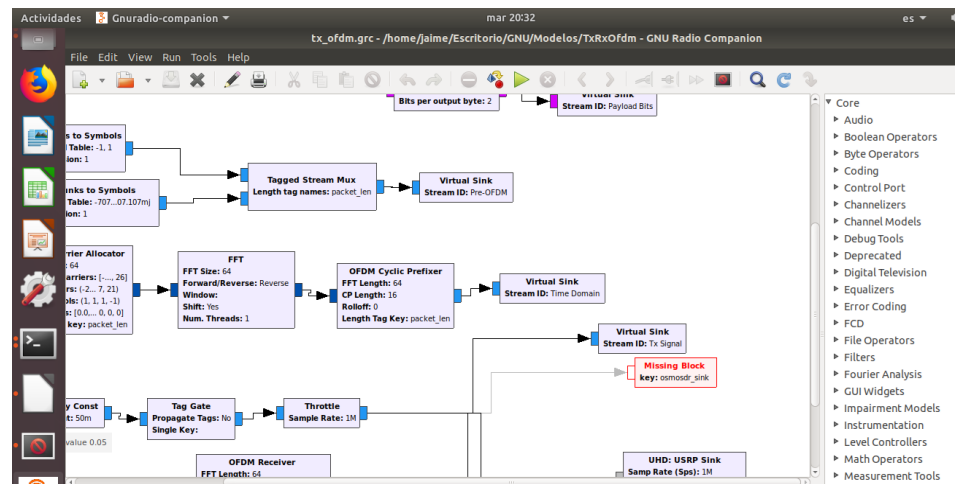
Ingresar los parámetros de transmisión y recepción de una señal en el sistema OFDM





Figuras 87, 88, 89 y 90 Ingreso de parámetros de transmisión-recepción de una señal en el sistema OFDM

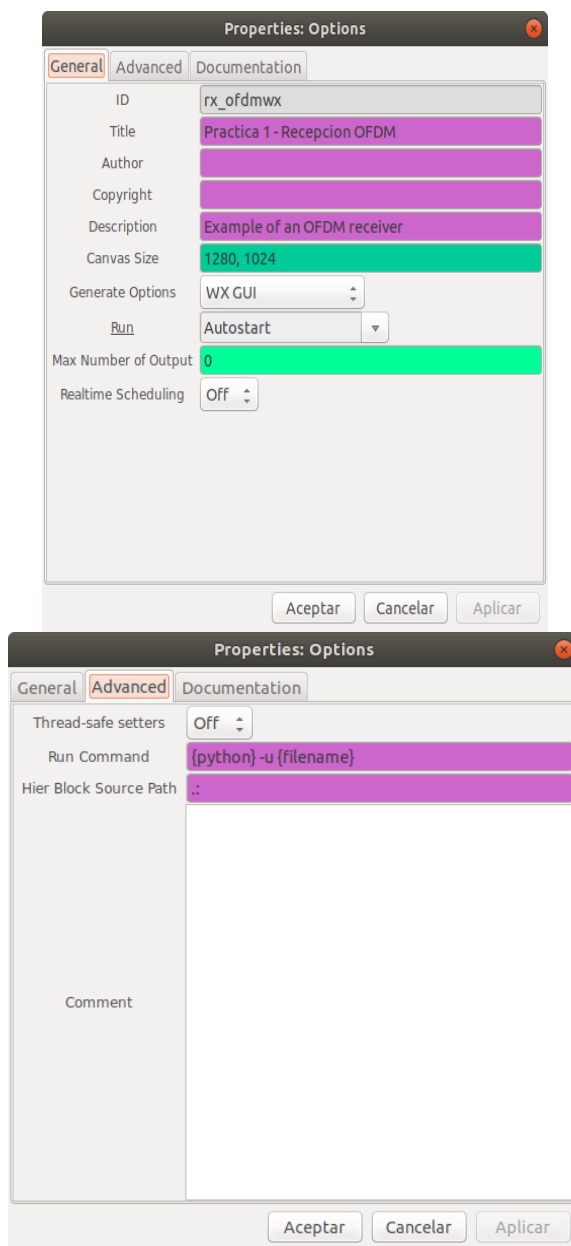
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho



Figuras 91 y 92 Parámetros de transmisión-recepción de señal en el sistema OFDM

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetro bloque Options

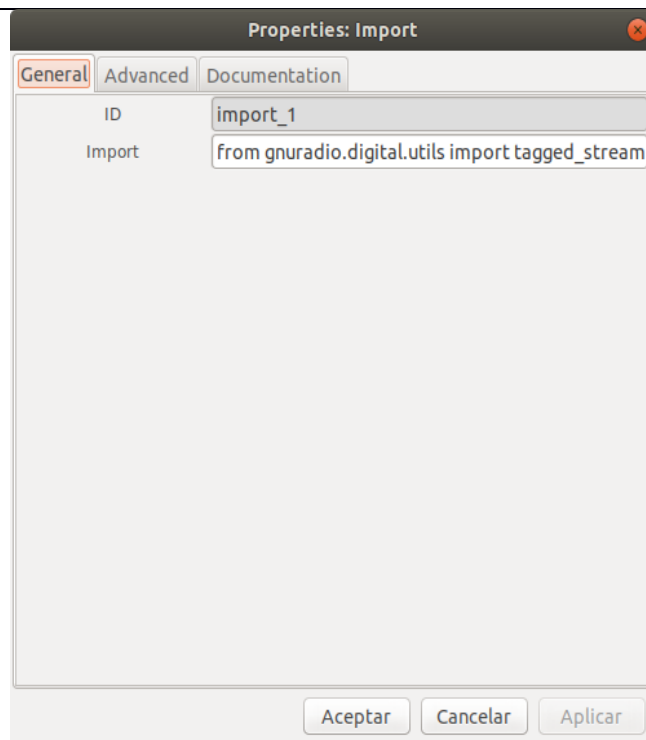


Figuras 93 y 94 Parámetros bloque Options

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Import

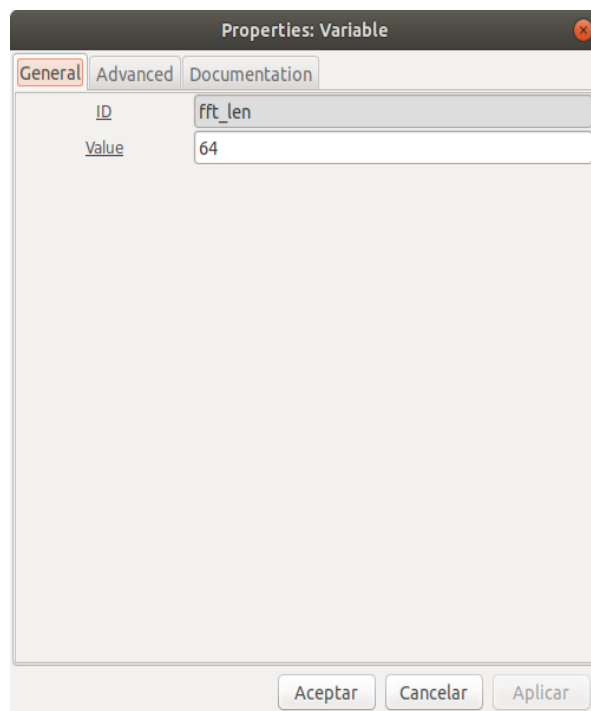
Importa módulos adicionales de python dentro del espacio de trabajo.

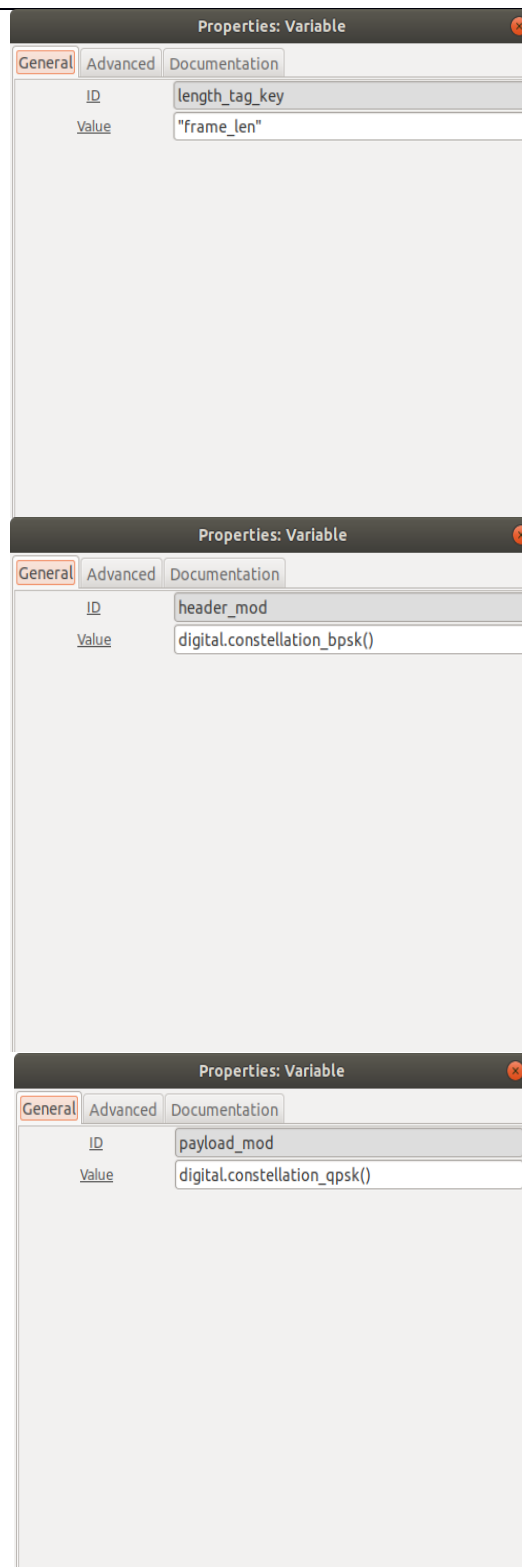


Figuras 95 Bloque Import

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Variables a crear





Figuras 96, 97, 98, 99 Bloque Import: Creación de variables
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

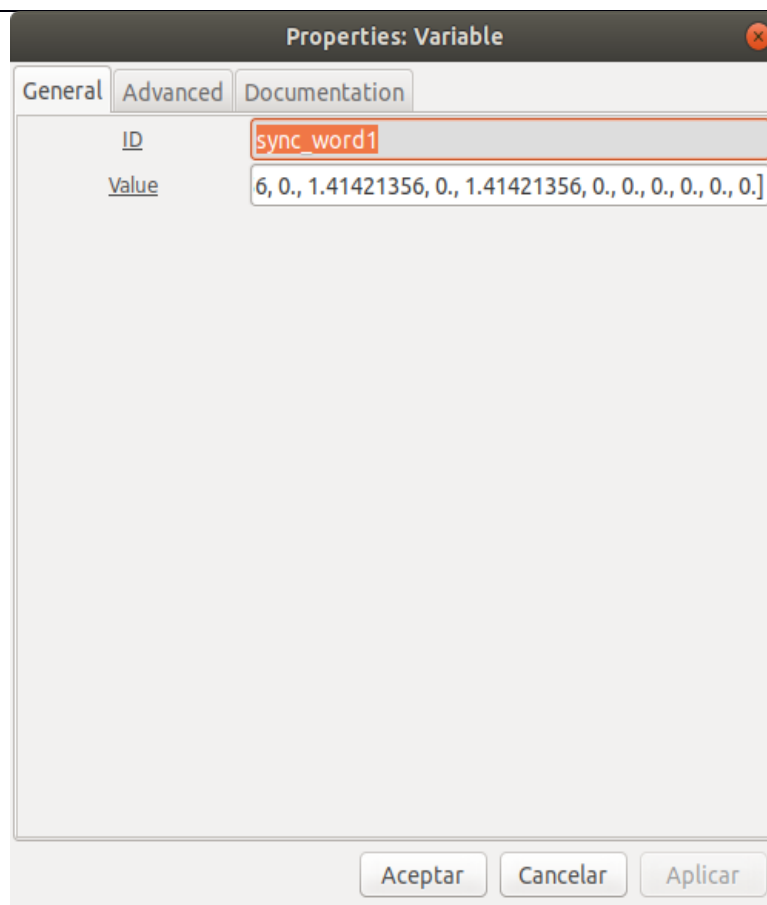


Figura 100 Ingreso de sync_word1

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor del value: [0., 0., 0., 0., 0., 0., 0., 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 0., 0., 0., 0., 0.]

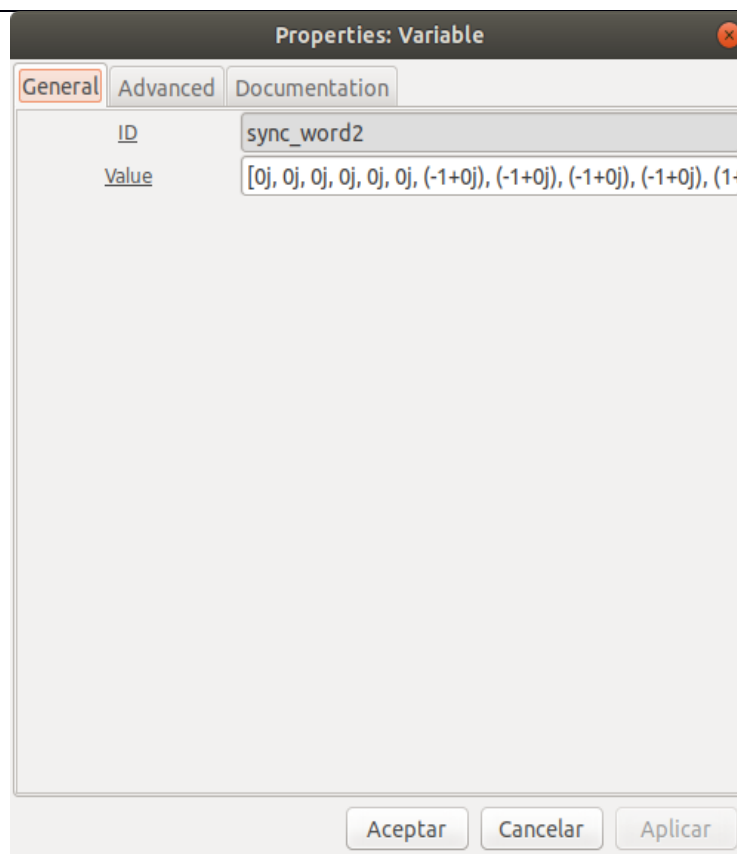


Figura 101 Ingreso de sync_word2

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor del value: [0j, 0j, 0j, 0j, 0j, 0j, (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), 0j, (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), 0j, 0j, 0j, 0j, 0j]

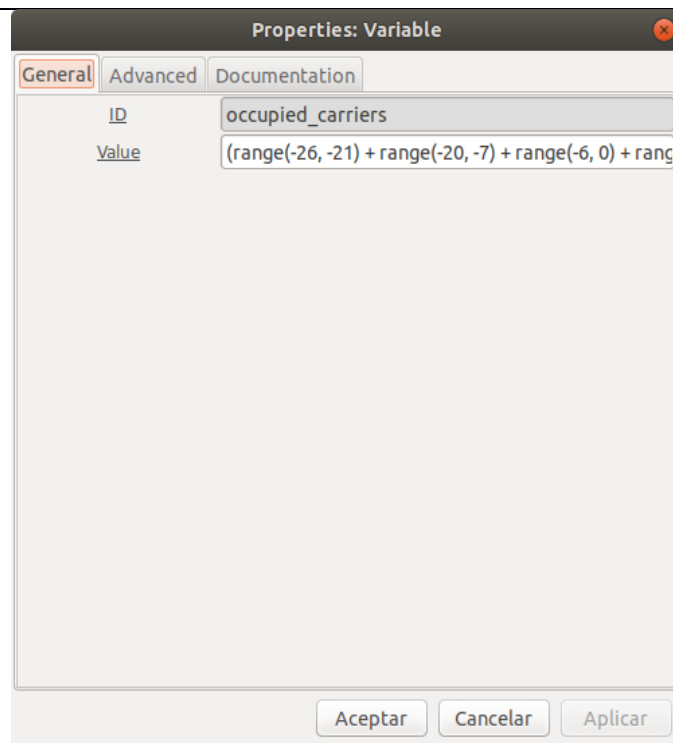


Figura 102 Ingreso de occupied_carriers

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor del value: (range(-26, -21) + range(-20, -7) + range(-6, 0) + range(1, 7) + range(8, 21) + range(22, 27),)

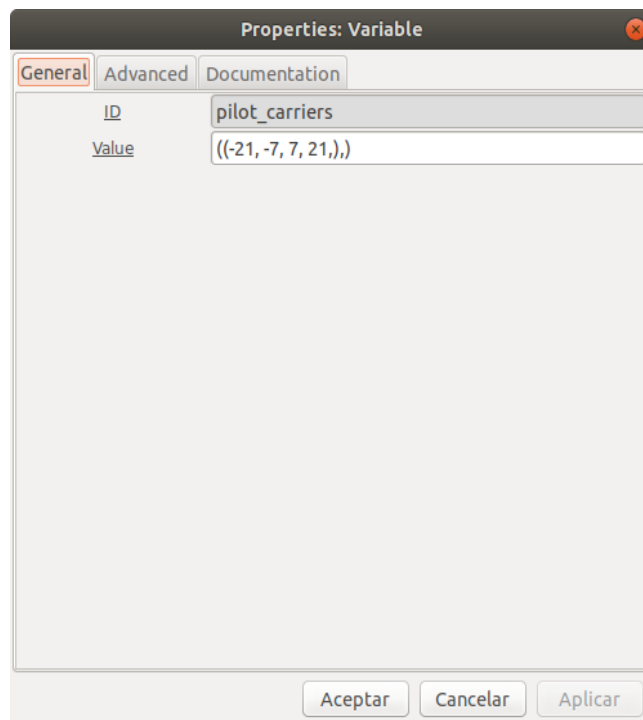


Figura 103 Ingreso de pilot_carriers

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor del value: digital.ofdm_equalizer_simplifiedfe(fft_len, header_mod.base(), occupied_carriers, pilot_carriers, pilot_symbols)

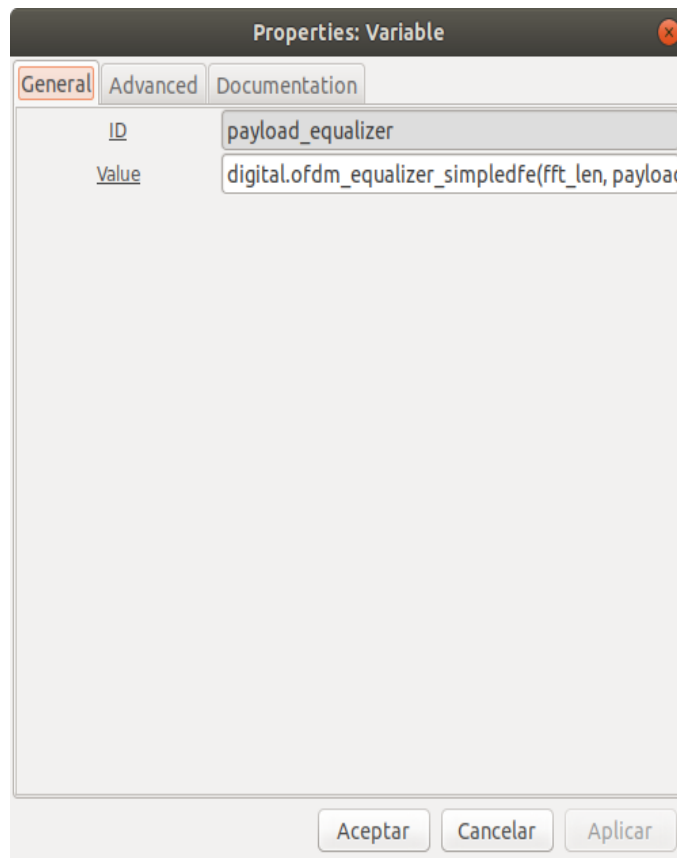


Figura 104 Valor del value

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WX GUI Text Box

Este bloque crea una variable con una caja de texto. Deja la etiqueta en blanco para usar la variable id como la etiqueta.

El formato debe ser una función lambda que convierte un valor en una cadena o ninguno para el formato por defecto.

Usar la opción de posición de la cuadrícula (row, column, row span, column span) para posicionar el gráfico.

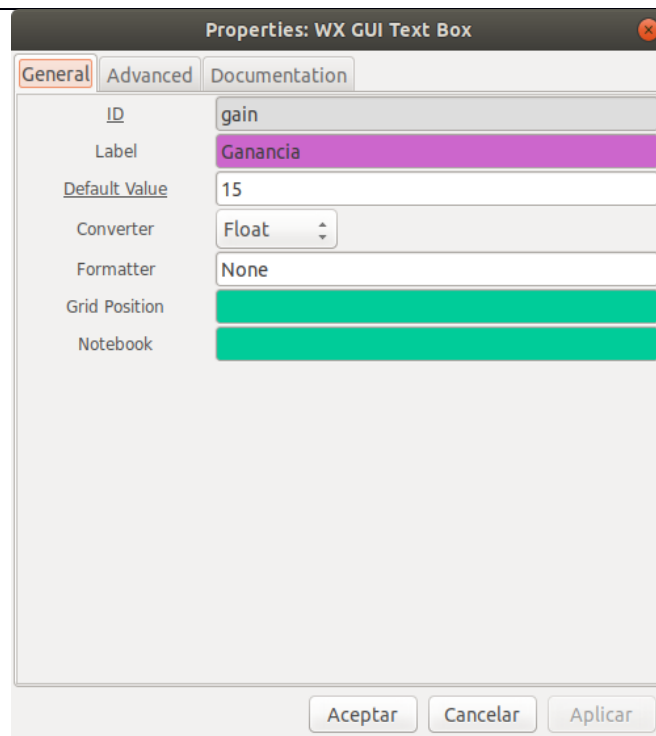


Figura 105 Propiedades WX GUI: Label ganancia
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

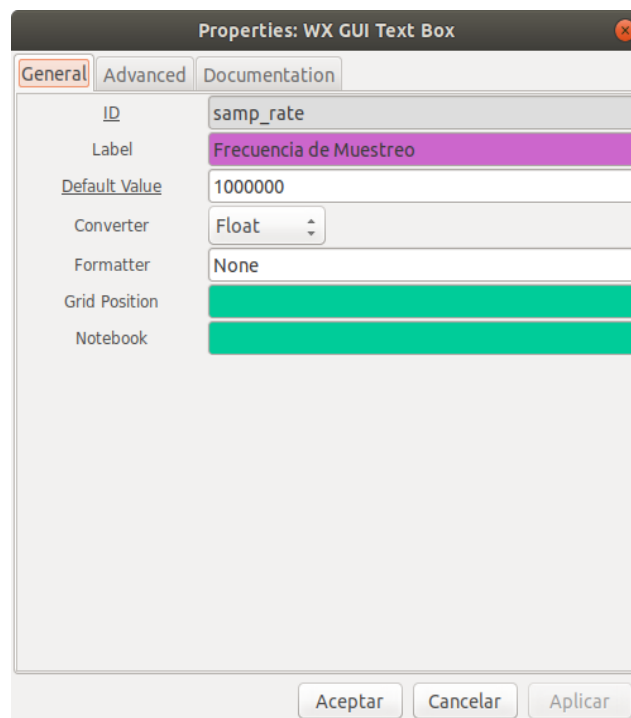


Figura 106 Propiedades WX GUI: Label frecuencia del muestreo
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetros UHD: USRP Source

The screenshot shows the 'Properties: UHD: USRP Source' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'RF Options', 'FE Corrections', 'Advanced', and 'Documentation'. The 'General' tab contains the following fields:

Field	Value
ID	uhd_usrp_source_0
Output Type	Complex float32
Wire Format	Complex int8
Stream args	
Stream channels	
Device Address	
Device Arguments	
Sync	don't sync
Clock Rate (Hz)	Default
Num Mboards	1
Mb0: Clock Source	Default
Mb0: Time Source	Default
Mb0: Subdev Spec	
Num Channels	1
Samp Rate (Sps)	samp_rate

At the bottom of the dialog are three buttons: 'Aceptar', 'Cancelar', and 'Aplicar'.

Figura 107 Parámetros UHD, USRP Source: General
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

The screenshot shows the 'Properties: UHD: USRP Source' dialog box with the 'RF Options' tab selected. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'RF Options', 'FE Corrections', 'Advanced', and 'Documentation'. The 'RF Options' tab contains the following fields:

Field	Value
Ch0: Center Freq (Hz)	cent_freq
Ch0: Gain Value	gain
Ch0: Gain Type	Absolute (dB)
Ch0: Antenna	RX2
Ch0: Bandwidth (Hz)	0

At the bottom of the dialog are three buttons: 'Aceptar', 'Cancelar', and 'Aplicar'.

Figura 108 Parámetros UHD, USRP Source: Opciones RF
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WX GUI FFT Sink

Establece el alfa promedio en 0 para el ajuste automático.

Deja la ventana en blanco para el tamaño predeterminado; de lo contrario, ingrese una tupla de píxeles (ancho, alto).

Bloque Schmidl & Cox OFDM synch

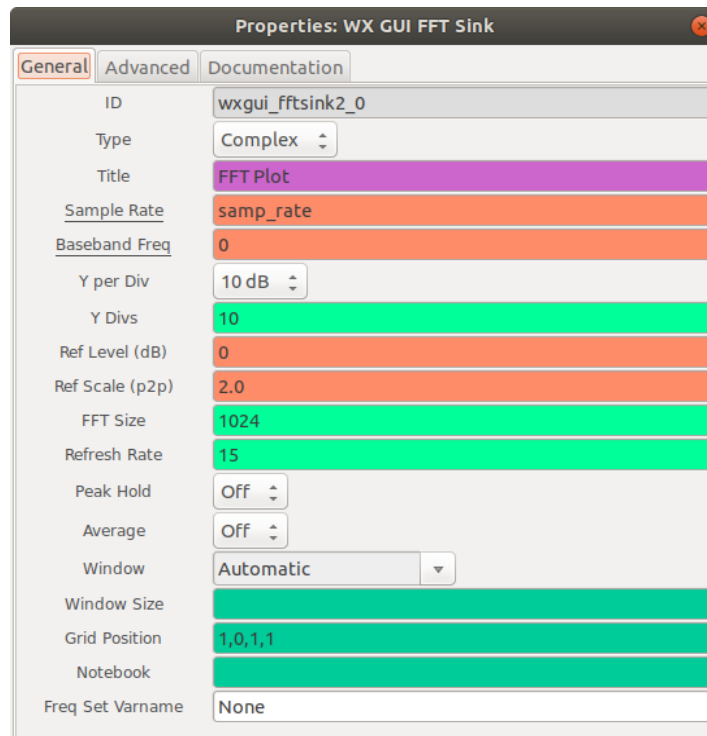


Figura 109 Propiedades WX GUI FFT Sink

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

La evaluación del desplazamiento de frecuencia gruesa se realiza en este bloque. Además, los taps iniciales del ecualizador no se calculan aquí.

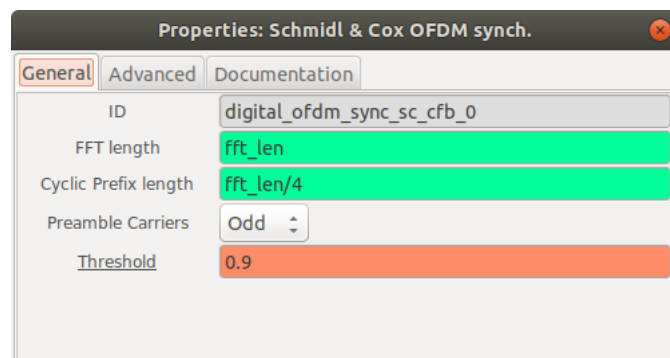


Figura 110 Taps iniciales del ecualizador

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Delay

Retrasa la entrada por un cierto número de muestras.

Los retrasos positivos insertan cero elementos al principio del flujo. Los retrasos negativos descartan elementos de la transmisión.

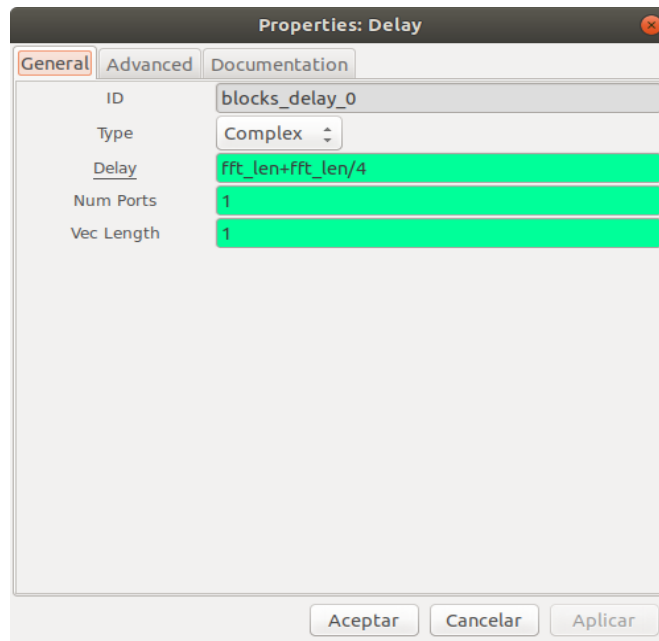


Figura 111 Bloque Delay

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Sin embargo, no puede inicializar este bloque con un retraso negativo. Eso lleva a un problema de causalidad con los buffers cuando se inicializan. Si necesita retrasar negativamente una ruta, entonces coloque el retraso positivo en la otra ruta.

Bloque Frequency Mod

Bloque modulador de la frecuencia

Toma una señal real de banda base ($x_m[n]$) y emite una señal de frecuencia modulada ($y[n]$) de acuerdo con:

Donde $x[n]$ es la muestra de entrada en el tiempo n y es la desviación de frecuencia. Los valores comunes son 5 kHz para canales FM de banda estrecha, como los sistemas de voz, y 75 KHz para FM de banda ancha, como las estaciones de FM de transmisión de audio.

En este bloque, el argumento de entrada no es la desviación de frecuencia. La sensibilidad especifica cuánto cambia la fase en función de la nueva muestra de entrada.

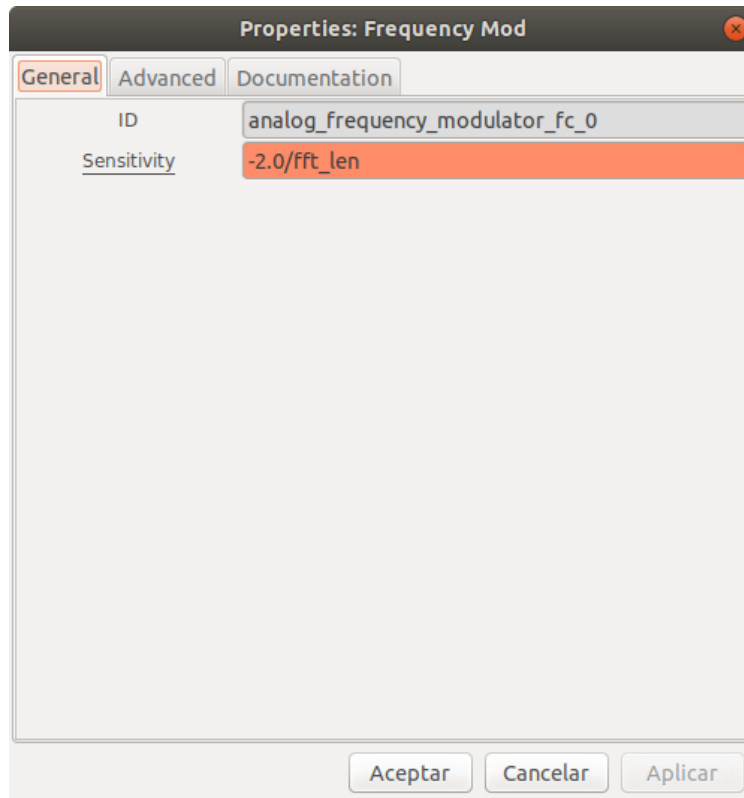


Figura 112 Bloque modulador de la frecuencia

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WX GUI Slider

Este bloque crea una variable con un control deslizante. Deje la etiqueta en blanco para usar la identificación de la variable como la etiqueta. El valor debe ser un número real. El valor debe estar entre el mínimo y el máximo. El número de pasos debe estar entre 0 y 1000.

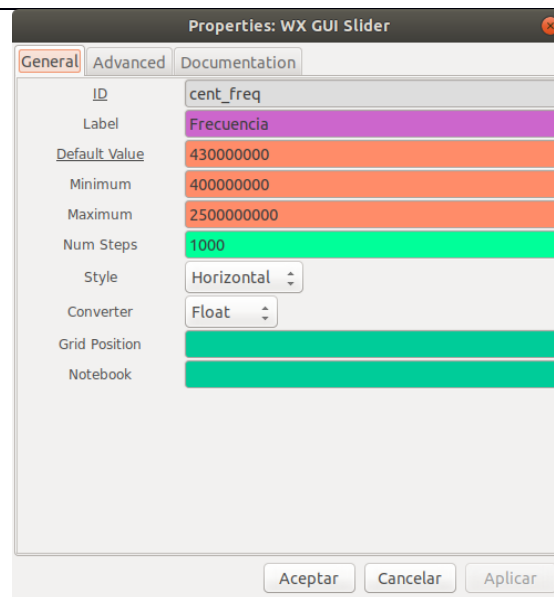


Figura 113 Bloque WX GUI Slider

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Multiply

Multiplica a través de todos los flujos de entrada.

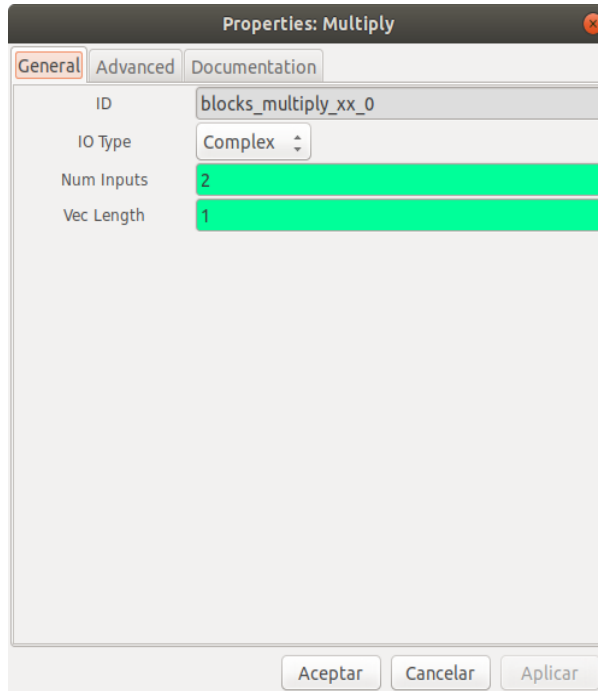


Figura 114 Bloque Multiply

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WX GUI Constellation Sink

Deja la ventana en blanco por el tamaño por defecto, en otro caso entra una tupla de píxeles (alto, ancho).

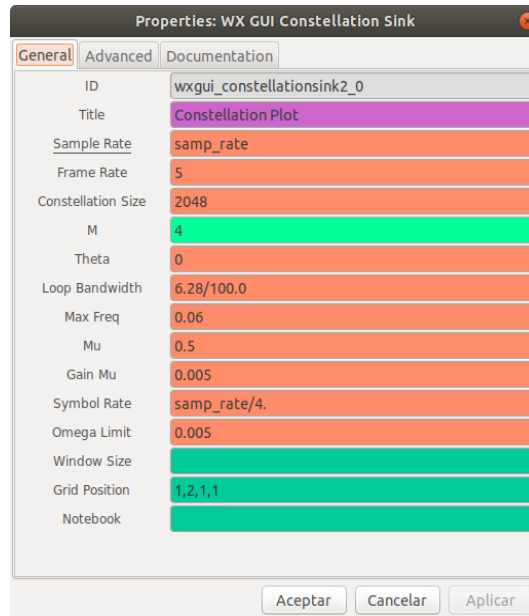


Figura 115 Bloque WX GUI Constellation Sink
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WX GUI Waterfall Sink

Establece el alfa promedio en 0 para el ajuste automático. Deja la ventana en blanco para el tamaño predeterminado; de lo contrario, ingrese una tupla de píxeles.

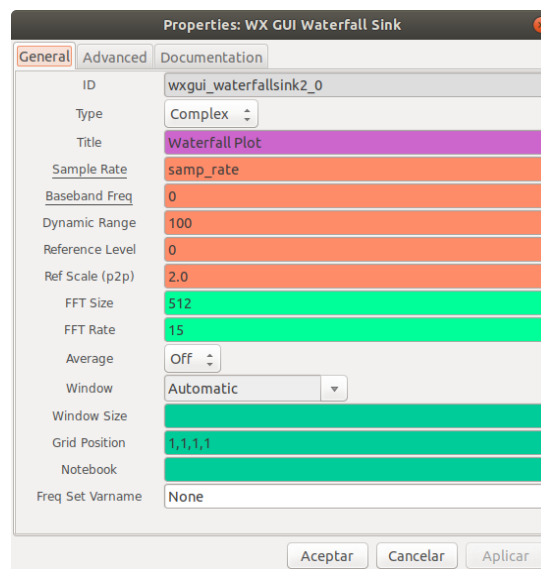


Figura 116 WX GUI Waterfall Sink
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Header/Payload Demux (HPD).

Este bloque está diseñado para demultiplexar paquetes de una transmisión en ráfaga. La aplicación típica para este bloque es el caso cuando está recibiendo paquetes con una longitud aún por determinar. Este bloque pasará la sección del encabezado a otros bloques para la demodulación. Usando la información del encabezado de modulado, luego emitirá la carga útil. El inicio del encabezado debe identificarse mediante una señal de activación.

Teoría de operación

Entrada 0 toma una transmisión continua de muestras (artículos). La entrada 1 es una entrada opcional para la señal de activación (marca de inicio de paquetes). En este caso, un valor distinto de cero en la entrada 1 identifica el comienzo de un paquete. De lo contrario, una etiqueta con la clave especificada en se utiliza como un activador (su valor es irrelevante).

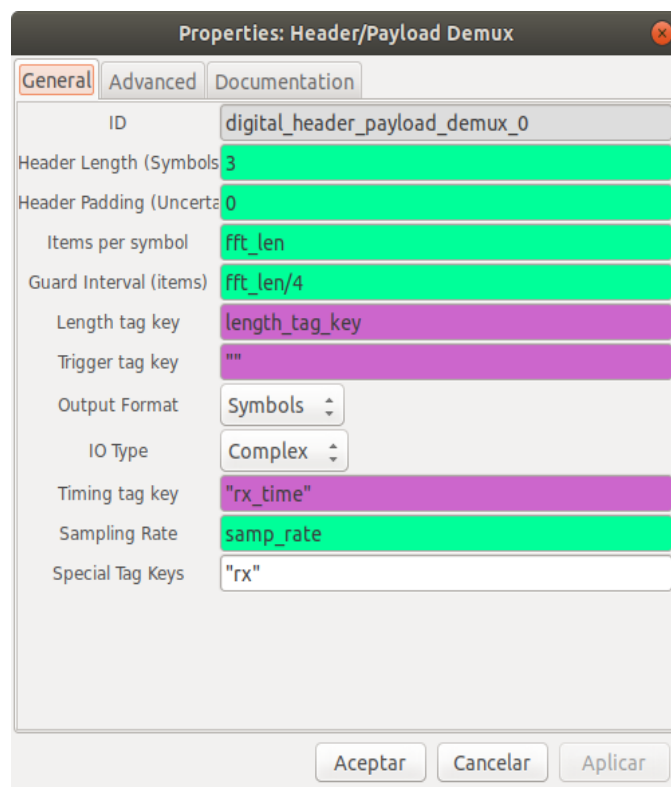


Figura 117 Bloque Header/Payload Demux

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Hasta que se detecta una señal de activación, todas las muestras se caen al suelo. Una vez que se detecta un disparador, se copia un total de elementos a la salida 0. El bloque se detiene hasta que recibe un mensaje en el puerto de mensajes. El mensaje debe ser un diccionario PMT; todos

los pares clave / valor se copian como etiquetas al primer elemento de la carga útil (que se supone que es el primer elemento después del encabezado). El valor correspondiente a la clave especificada en se lee y se toma como la longitud de la carga útil. La carga útil, junto con los datos del encabezado como etiquetas, se copia a la salida 1.

Si la demodulación del encabezado falla, el encabezado debe enviar un PMT con el valor `pmt :: PMT_F`. El estado se restablece y el encabezado se ignora.

Bloque Constellation Decoder

Decodifica un punto de constelación desde un espacio complejo hacia bits basados en el mapeo del objeto.

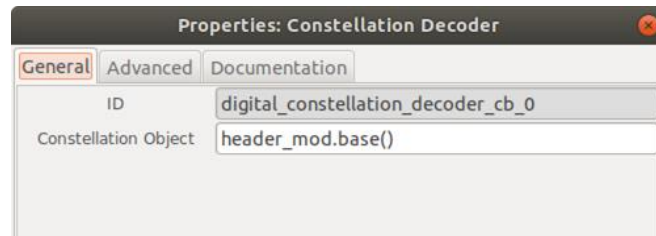


Figura 118 Bloque Constellation Decoder

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque FFT

Transformada de Fourier

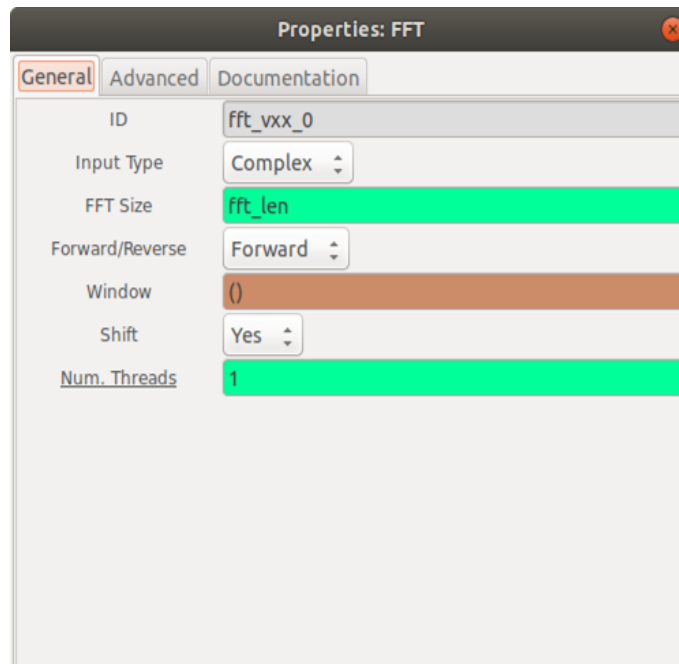


Figura 119 Bloque FFT

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Channel Estimation

Estimación del canal y del desplazamiento de la frecuencia aproximada para OFDM a partir de preámbulos

Entrada: símbolos OFDM (en el dominio de la frecuencia). Se espera que los primeros (o dos) símbolos sean símbolos de sincronización, que se utilizan para estimar el desplazamiento de frecuencia aproximado y los toques iniciales del ecualizador (estos símbolos se eliminan de la secuencia). Lo siguiente se pasa sin modificar (la ecualización real debe realizarse en otro lugar).

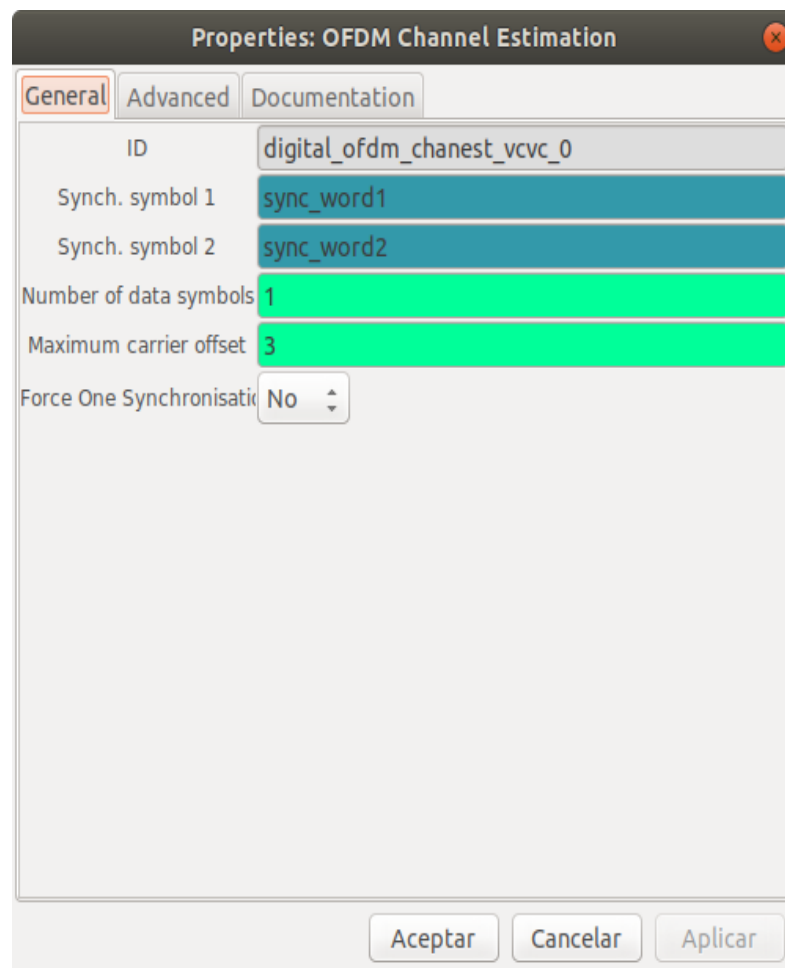


Figura 120 Propiedades canal OFDM

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Frame Equalizer

Ecualizador de cuadros OFDM.

Realiza la ecualización en una o dos dimensiones en un marco OFDM etiquetado.

Esto hace dos cosas: primero, elimina el desplazamiento de portador grueso. Si se encuentra una etiqueta en el primer elemento con la clave 'ofdm_sync_carr_offset', esto se interpreta como el desplazamiento de frecuencia aproximado en número de portadoras. A continuación, realiza la ecualización en una o dos dimensiones en un marco OFDM etiquetado. La ecualización real se realiza mediante un objeto `ofdm_frame_equalizer`, fuera del bloque.

Tenga en cuenta que la etiqueta con el desplazamiento de portador grueso no se elimina. Los bloques aguas abajo de este bloque no deben intentar corregir también este desplazamiento.

Entrada: una serie etiquetada de símbolos OFDM. Salida: igual que la entrada, pero ecualizada y con corrección de frecuencia.

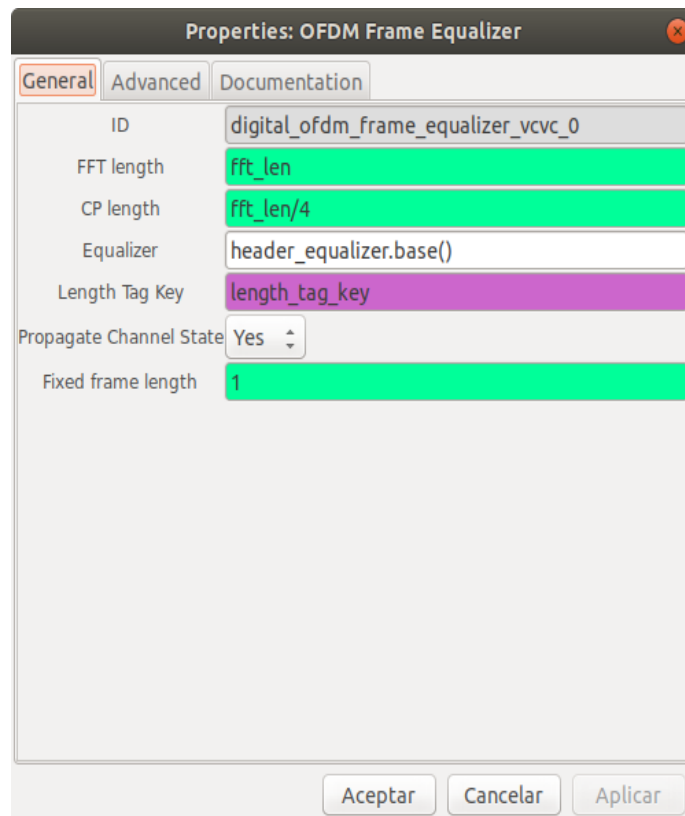


Figura 121 Corrección de frecuencia

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque OFDM Serializer

Serializa símbolos de modulación complejos de subportadoras OFDM.

Este es el bloque inverso al `carrier_allocator_cvc`. Envía los símbolos de datos complejos como un flujo etiquetado, descartando los símbolos piloto.

Si se da, se analizan dos etiquetas diferentes: La primera clave () especifica el número de símbolos OFDM en el marco en la entrada. La segunda clave () especifica el número de símbolos complejos que se codifican en este marco. Si se da, esta segunda clave se usa en la salida, de lo contrario, se usa. Si se dan ambos, la longitud del paquete especifica el número máximo de elementos de salida, y la longitud del marco especifica el número exacto de elementos de entrada consumidos.

Es posible corregir un desplazamiento de portadora en esta función pasando otra etiqueta con dicho desplazamiento.

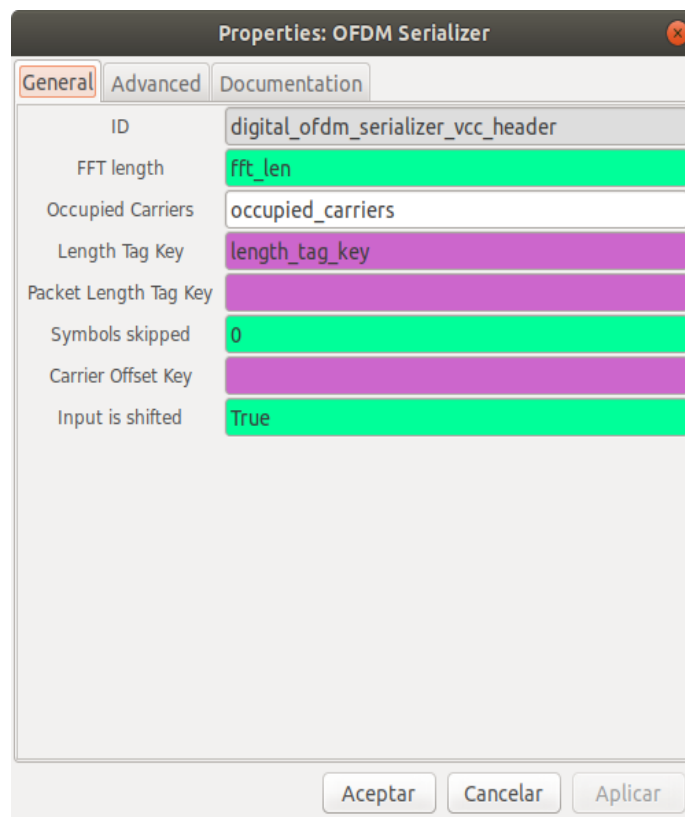


Figura 122 Serializador OFDM

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

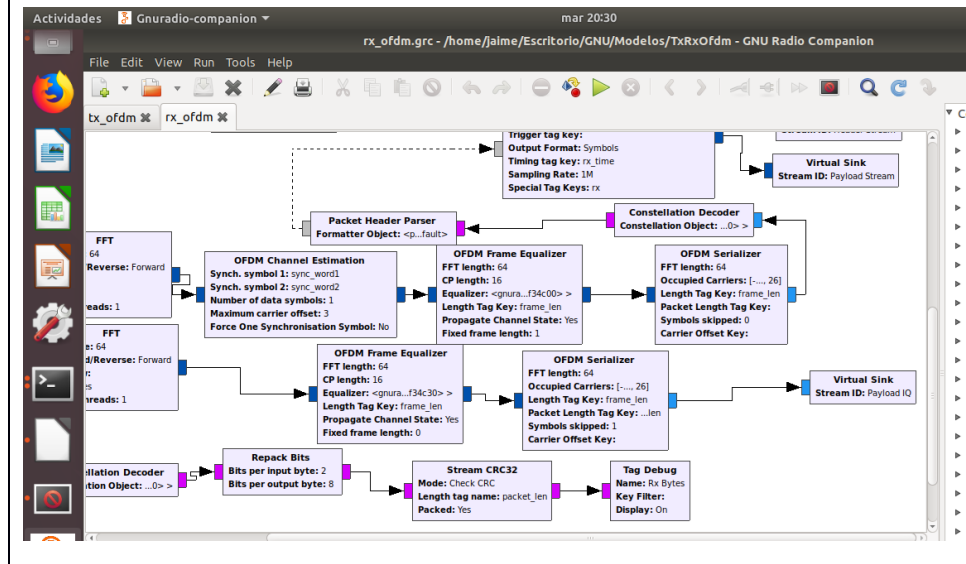
Paso 3

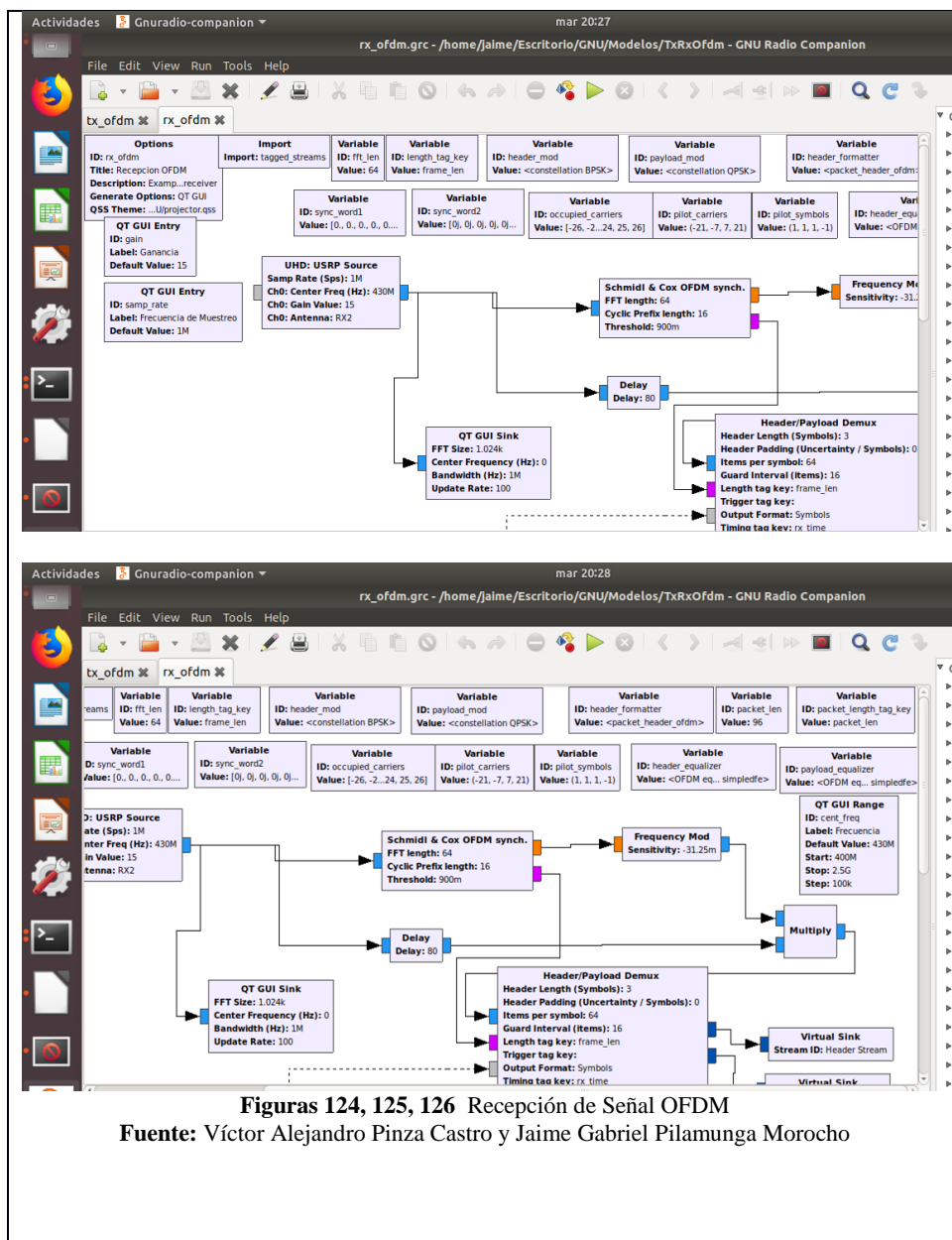
Ejecutamos nuestra aplicación

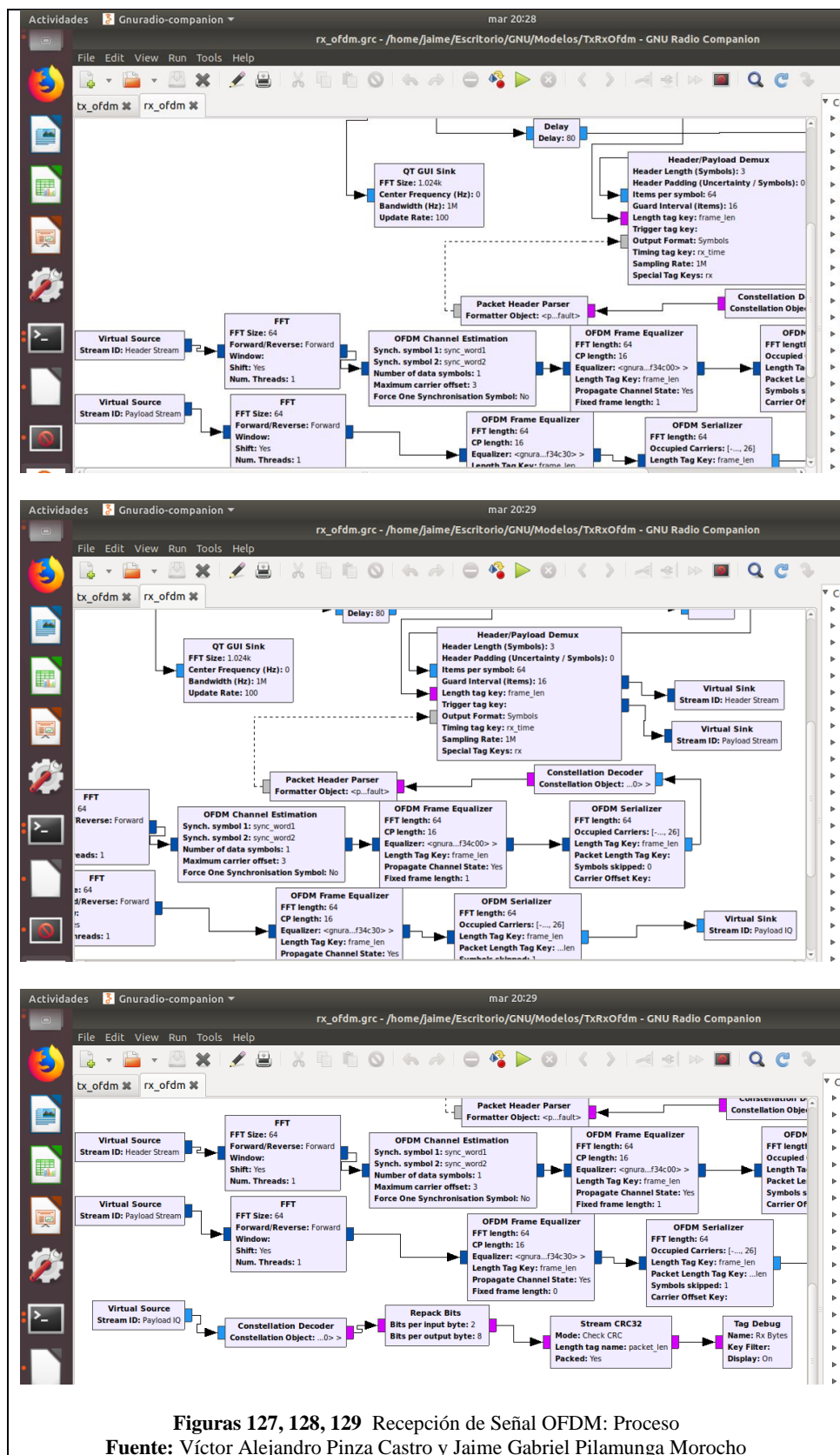


Figura 123 Aplicación ejecutándose
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Recepción de Señal OFDM







Figuras 127, 128, 129 Recepción de Señal OFDM: Proceso
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

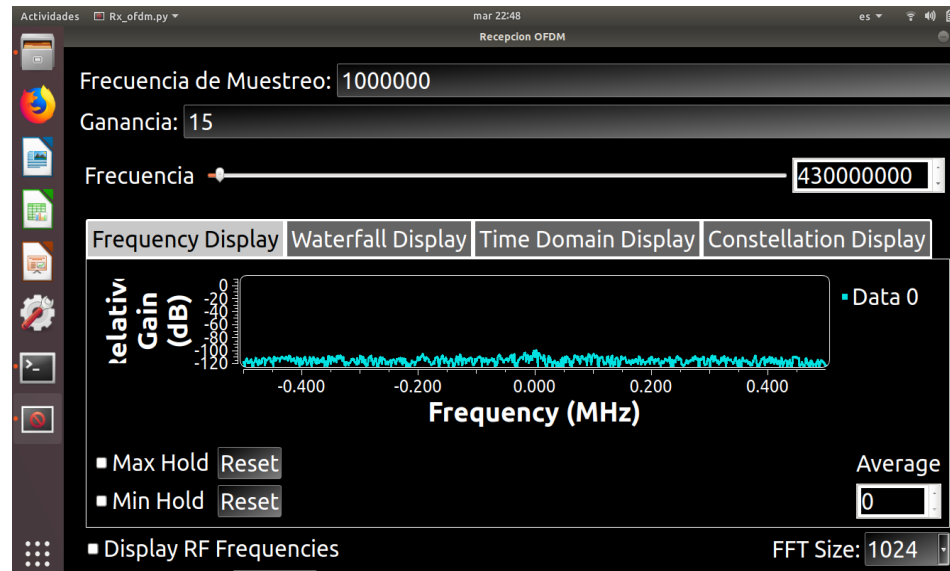


Figura 130 Recepción de Señal OFDM: Frecuencia de muestreo
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho


CONCLUSIONES

- ✓ La naturaleza "ortogonal" de la multiplexación de frecuencia, seguida del prefijo elimina efectivamente ISI en canales de ruta única y lo reduce considerablemente en canales de ruta múltiple.
- ✓ El esquema de prefijo cíclico y DFT permiten una señal rápida y fácil procesamiento, esto sucede en el hardware y en el software, lo que permite una fácil estimación y ecualización del canal, a través de símbolos fijos conocidos (símbolos piloto).
- ✓ OFDM puede admitir el acceso dinámico a paquetes. La codificación espacio-temporal pueden realizarse en OFDM y todos los beneficios se pueden obtener fácilmente. Modulación adaptativa y tono/potencia. La asignación también se puede realizar en OFDM.

RECOMENDACIONES

- ✓ OFDM es altamente sensible a los errores de sincronización de tiempo y frecuencia, especialmente con los errores de sincronización de frecuencia.
- ✓ La demodulación de una señal OFDM con un desplazamiento en la frecuencia puede conducir a una alta tasa de error de bits.

PRÁCTICA 8.

 GUÍA DE PRÁCTICA DE LABORATORIO		
CARRERA: Ingeniería Electrónica		ASIGNATURA:
NRO. PRÁCTICA:	8	TÍTULO PRÁCTICA: Wifi
OBJETIVO GENERAL: Estudiar el comportamiento de una señal Wifi, a través del procesamiento de señales en tiempo real para uso del sistema SDR. OBJETIVOS ESPECIFICOS: ➤ Analizar el encabezado MAC y extraer la carga útil de las redes IEEE 802.11a / g / p.		
INSTRUCCIONES		
Introducción La investigación experimental en protocolos de comunicación inalámbrica frecuentemente requiere acceso completo a todas las capas de protocolo, debajo o hasta, e incluyendo la capa física. Software definido Plataformas de hardware (SDR), junto con señal de tiempo real en marcos de procesamiento, ofrecen una base para implementar que puedan permitir tal experimentación y sofisticación de mediciones. Presentamos una frecuencia ortogonal completa. Receptor de multiplexación por división (OFDM) implementado en GNU Radio y equipado para funcionar con un Ettus USRP B205 mini-i. Para mejora de nuestro conocimiento, este es el primer prototipo de un receptor OFDM basado en GNU Radio para esta tecnología. Nuestro receptor comprende todas las capas hasta analizar la MAC encabezado y extracción de la carga útil de IEEE 802.11a / g / p net-trabajos.		

Es compatible con WiFi teniendo 20 MHz de ancho de banda y un ancho de banda de 10 MHz con IEEE 802.11p DSRC.

Ofrecemos un sistema de fácil acceso que se puede utilizar fácilmente.

Para experimentar con nuevos algoritmos de procesamiento de señales.

Actividades por desarrollar

Paso 1

Instalar las siguientes dependencias

Swig

Swig es requerida para crear los enlaces de phyton.

En nuestro terminal escribimos lo siguiente:

```
sudo apt-get install swig
```

```
sudo port install swig
```

```
log4cpp
```

Usamos la nueva característica de logging de GNU Radio, la cual se llama log4cpp. Esta debe ser una dependencia opcional algún día, pero recientemente es requerida. La instalamos así:

```
sudo apt-get install liblog4cpp5-dev
```

```
sudo port install log4cpp
```

gr-foo

Se necesita el repositorio gr-foo. Por ejemplo, el conector Wireshark.

Encontramos estos bloques en la siguiente dirección

<https://github.com/bastibl/gr-foo>.

Los instalamos de la siguiente manera:

```
git clone https://github.com/bastibl/gr-foo.git
```

```
cd gr-foo
```

```
mkdir build
```

```
cd build
```


cmake ..

make

sudo make install

sudo ldconfig

Instalación de la librería de gr-ieee802-11

Para instalar los bloques, desde nuestro terminal ejecutamos:

git clone git://github.com/bastibl/gr-ieee802-11.git

cd gr-ieee802-11

mkdir build

cd build

cmake ..

make

sudo make install

sudo ldconfig

Paso 2

Crear grc de receptor WiFi

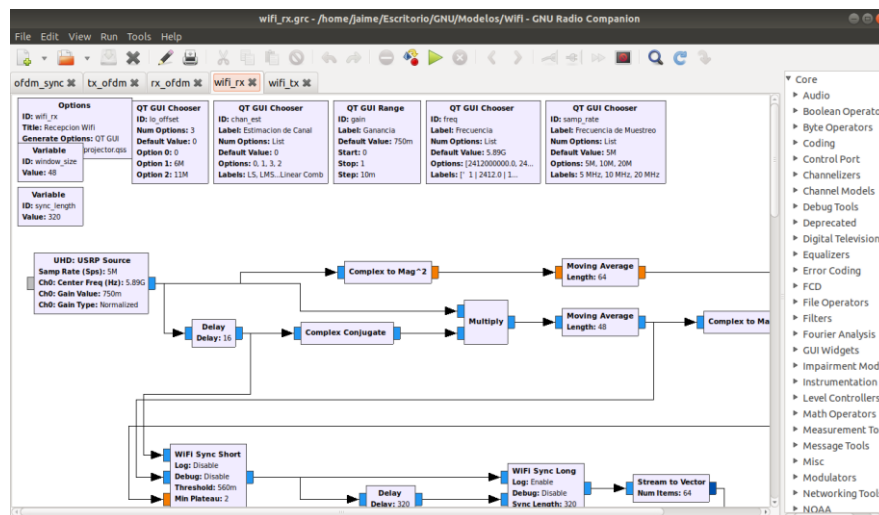


Figura 131 Creación de grc de receptor WiFi
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

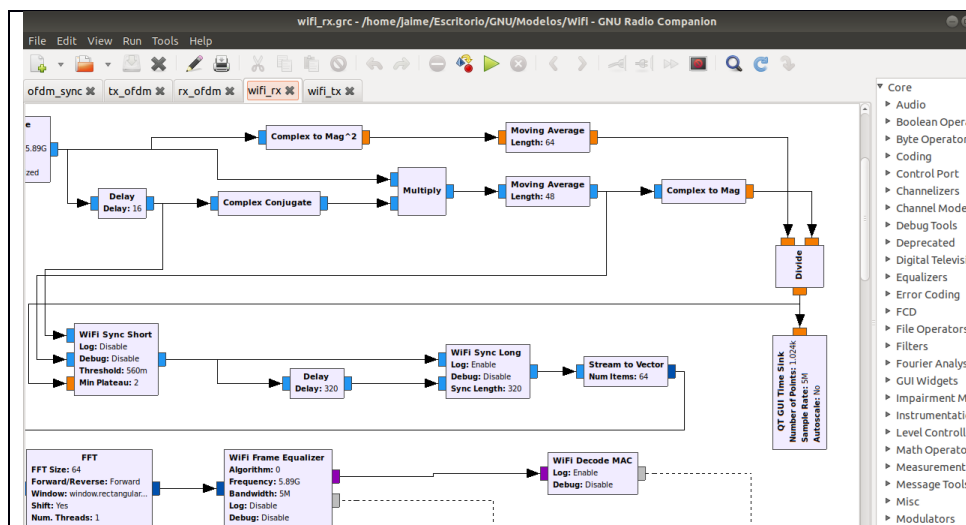


Figura 132 Grc de receptor WiFi

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque QT GUI Chooser

Este bloque crea una variable con opciones enumeradas. El widget de GUI se implementa como un cuadro combinado o un grupo de botones de radio. Deja la etiqueta en blanco para usar la identificación de la variable como la etiqueta. Elige el número de opciones disponibles para su seleccionador.

Cuando la etiqueta se deja vacía, la opción es escogida como etiqueta. Establece la cantidad de opciones en "lista" para ingresar una lista única de opciones y etiquetas. Cuando las etiquetas están en una lista vacía, las opciones se utilizarán como etiqueta.

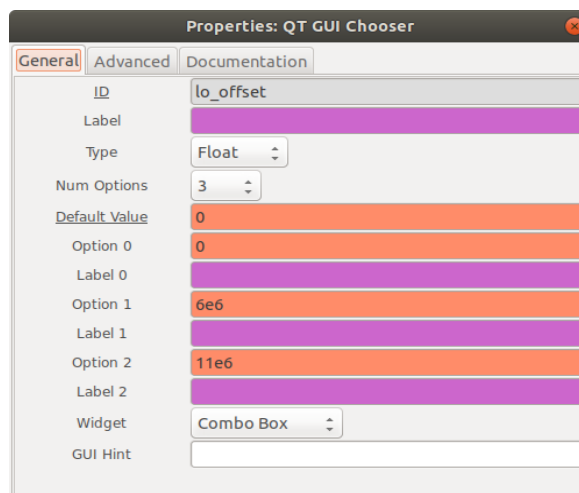


Figura 133 Bloque QT GUI Chooser

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetros de bloque QT GUI Chooser

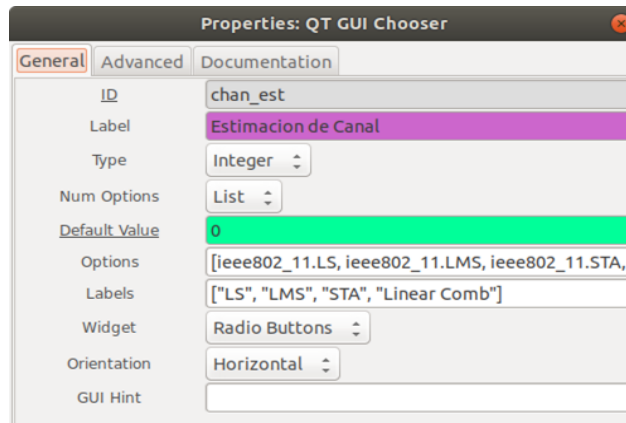


Figura 134 Parámetros de bloque QT GUI Chooser

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

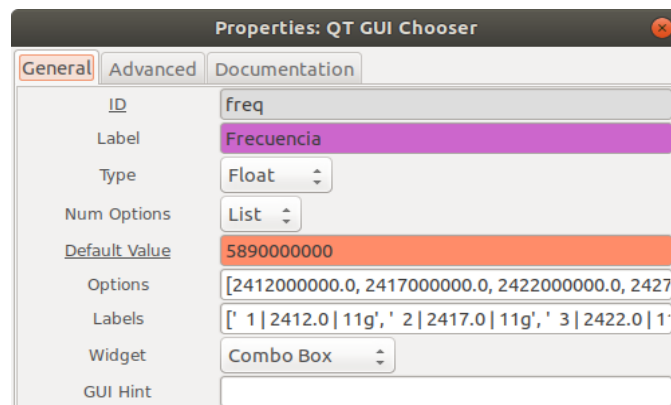


Figura 135 Parámetros de bloque QT GUI Chooser: Valor de options

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor de options: [2412000000.0, 2417000000.0, 2422000000.0, 2427000000.0, 2432000000.0, 2437000000.0, 2442000000.0, 2447000000.0, 2452000000.0, 2457000000.0, 2462000000.0, 2467000000.0, 2472000000.0, 2484000000.0, 5170000000.0, 5180000000.0, 5190000000.0, 5200000000.0, 5210000000.0, 5220000000.0, 5230000000.0, 5240000000.0, 5250000000.0, 5260000000.0, 5270000000.0, 5280000000.0, 5290000000.0, 5300000000.0, 5310000000.0, 5320000000.0, 5500000000.0, 5510000000.0, 5520000000.0, 5530000000.0, 5540000000.0, 5550000000.0, 5560000000.0, 5570000000.0, 5580000000.0, 5590000000.0, 5600000000.0, 5610000000.0, 5620000000.0, 5630000000.0, 5640000000.0, 5660000000.0, 5670000000.0, 5680000000.0, 5690000000.0, 5700000000.0, 5710000000.0, 5720000000.0, 5745000000.0, 5755000000.0, 5765000000.0, 5775000000.0, 5785000000.0, 5795000000.0, 5805000000.0, 5825000000.0, 5860000000.0, 5870000000.0, 5880000000.0, 5890000000.0, 5900000000.0, 5910000000.0, 5920000000.0]

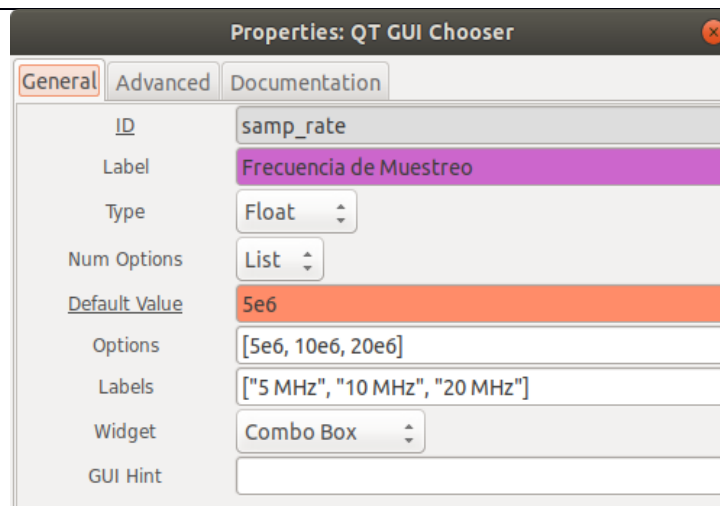


Figura 136 Parámetros de bloque QT GUI Chooser: Valor de labels
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Valor de Labels: [' 1 | 2412.0 | 11g', ' 2 | 2417.0 | 11g', ' 3 | 2422.0 | 11g', ' 4 | 2427.0 | 11g', ' 5 | 2432.0 | 11g', ' 6 | 2437.0 | 11g', ' 7 | 2442.0 | 11g', ' 8 | 2447.0 | 11g', ' 9 | 2452.0 | 11g', ' 10 | 2457.0 | 11g', ' 11 | 2462.0 | 11g', ' 12 | 2467.0 | 11g', ' 13 | 2472.0 | 11g', ' 14 | 2484.0 | 11g', ' 34 | 5170.0 | 11a', ' 36 | 5180.0 | 11a', ' 38 | 5190.0 | 11a', ' 40 | 5200.0 | 11a', ' 42 | 5210.0 | 11a', ' 44 | 5220.0 | 11a', ' 46 | 5230.0 | 11a', ' 48 | 5240.0 | 11a', ' 50 | 5250.0 | 11a', ' 52 | 5260.0 | 11a', ' 54 | 5270.0 | 11a', ' 56 | 5280.0 | 11a', ' 58 | 5290.0 | 11a', ' 60 | 5300.0 | 11a', ' 62 | 5310.0 | 11a', ' 64 | 5320.0 | 11a', ' 100 | 5500.0 | 11a', ' 102 | 5510.0 | 11a', ' 104 | 5520.0 | 11a', ' 106 | 5530.0 | 11a', ' 108 | 5540.0 | 11a', ' 110 | 5550.0 | 11a', ' 112 | 5560.0 | 11a', ' 114 | 5570.0 | 11a', ' 116 | 5580.0 | 11a', ' 118 | 5590.0 | 11a', ' 120 | 5600.0 | 11a', ' 122 | 5610.0 | 11a', ' 124 | 5620.0 | 11a', ' 126 | 5630.0 | 11a', ' 128 | 5640.0 | 11a', ' 132 | 5660.0 | 11a', ' 134 | 5670.0 | 11a', ' 136 | 5680.0 | 11a', ' 138 | 5690.0 | 11a', ' 140 | 5700.0 | 11a', ' 142 | 5710.0 | 11a', ' 144 | 5720.0 | 11a', ' 149 | 5745.0 | 11a (SRD)', ' 151 | 5755.0 | 11a (SRD)', ' 153 | 5765.0 | 11a (SRD)', ' 155 | 5775.0 | 11a (SRD)', ' 157 | 5785.0 | 11a (SRD)', ' 159 | 5795.0 | 11a (SRD)', ' 161 | 5805.0 | 11a (SRD)', ' 165 | 5825.0 | 11a (SRD)', ' 172 | 5860.0 | 11p', ' 174 | 5870.0 | 11p', ' 176 | 5880.0 | 11p', ' 178 | 5890.0 | 11p', ' 180 | 5900.0 | 11p', ' 182 | 5910.0 | 11p', ' 184 | 5920.0 | 11p']

Bloque QT GUI Range

Este bloque crea una variable con un control deslizante. Deje la etiqueta en blanco para usar la identificación de la variable como la etiqueta. El valor debe ser un número real. El valor debe estar entre el inicio y la parada.

La sugerencia de GUI se puede utilizar para colocar el widget dentro de la aplicación.

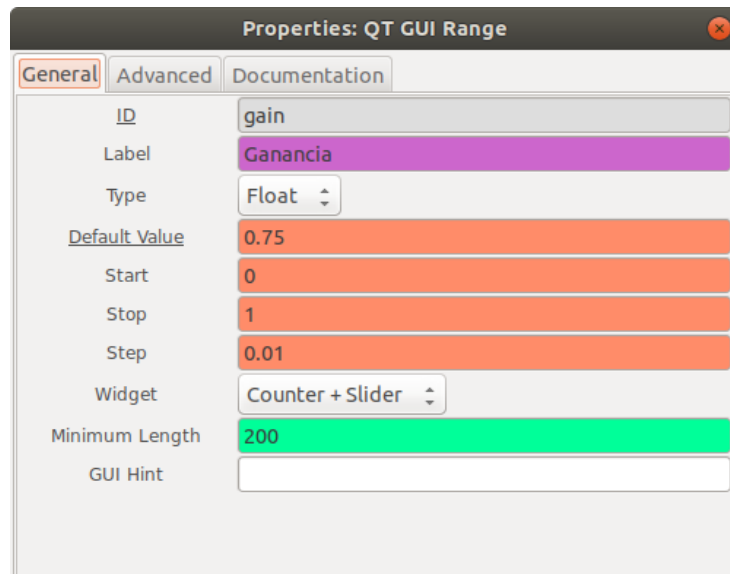


Figura 137 Bloque QT GUI Range

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque UHD USRP Source: parámetros

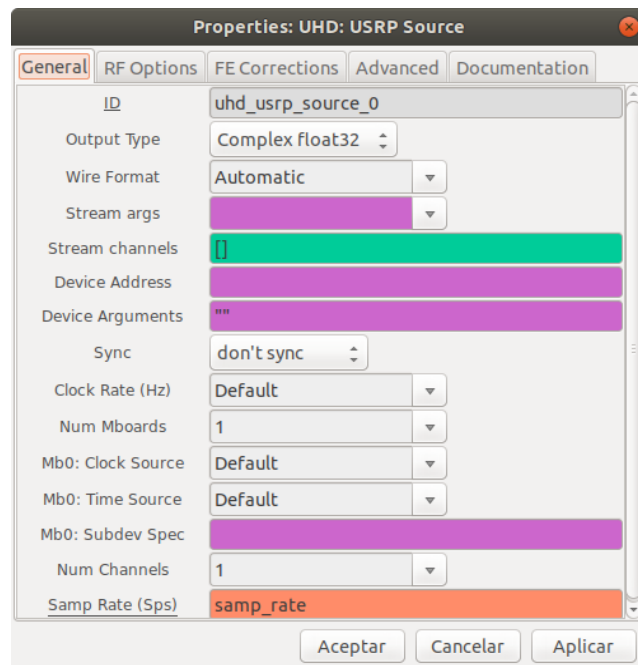


Figura 138 Bloque UHD USRP Source: parámetros

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WiFi Sync Short

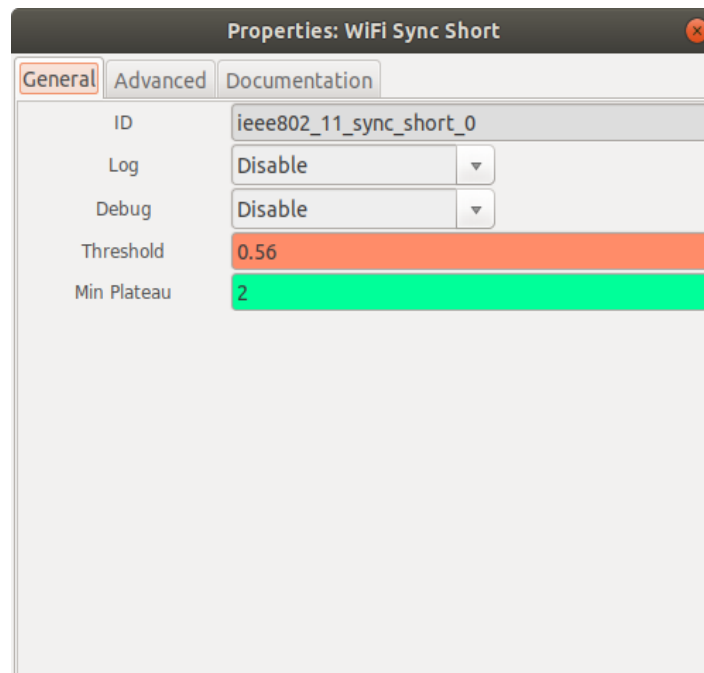


Figura 139 Bloque WiFi Sync Short

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Sincronización corta de WiFi

Bloque WiFi Sync Long

Sincronización larga Wifi

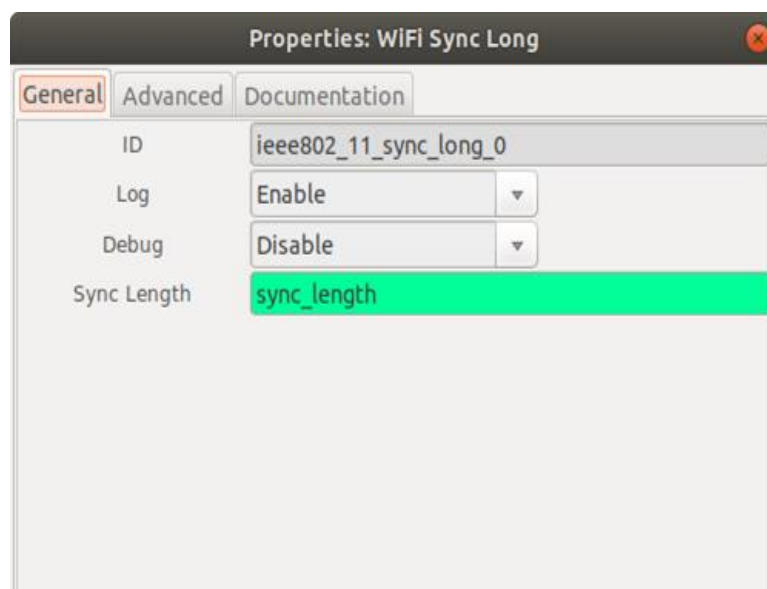


Figura 140 Bloque WiFi Sync Long

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

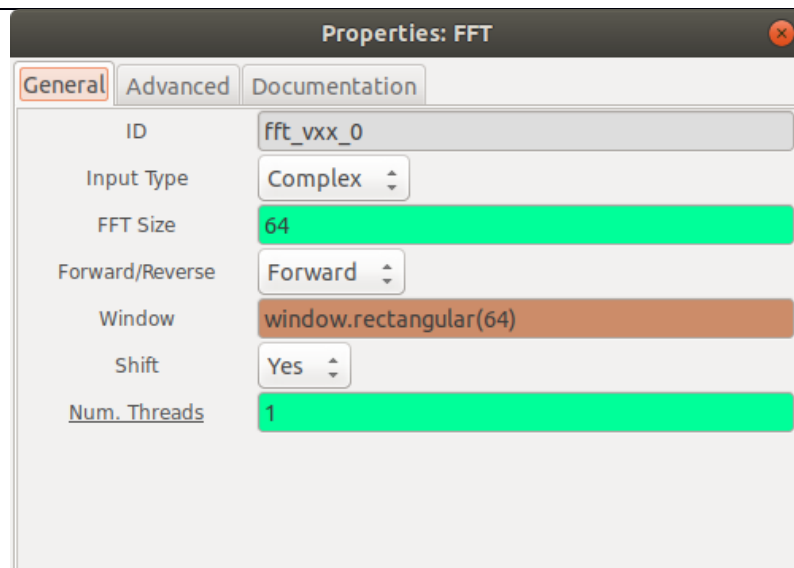


Figura 141 Sincronización larga Wifi

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Parámetros Bloque FFT

Bloque Wifi Frame Equalizer

Ecualizador de frame de WiFi

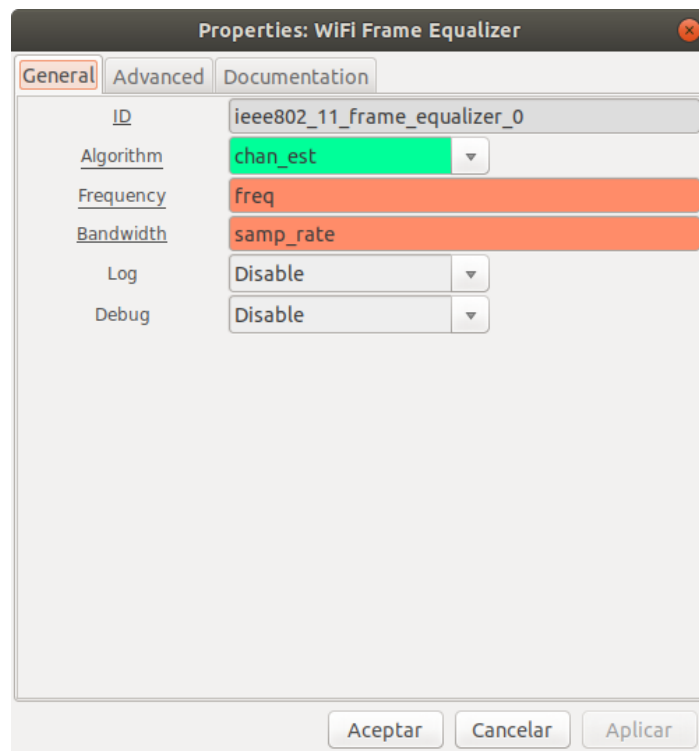


Figura 142 Ecualizador de frame de WiFi

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque WiFi Decode MAC

Decodificador WiFi MAC

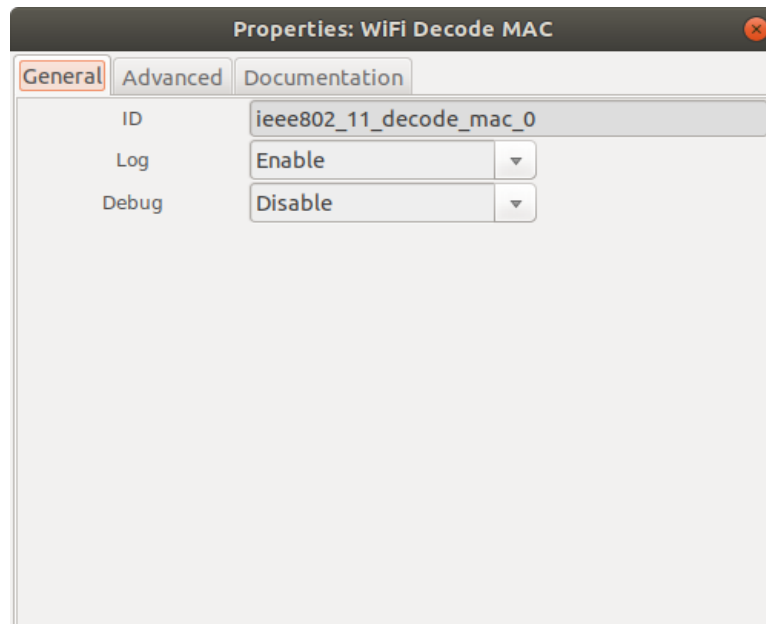


Figura 143 Decodificador WiFi MAC

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque Wireshark Connector

Conector Wireshark

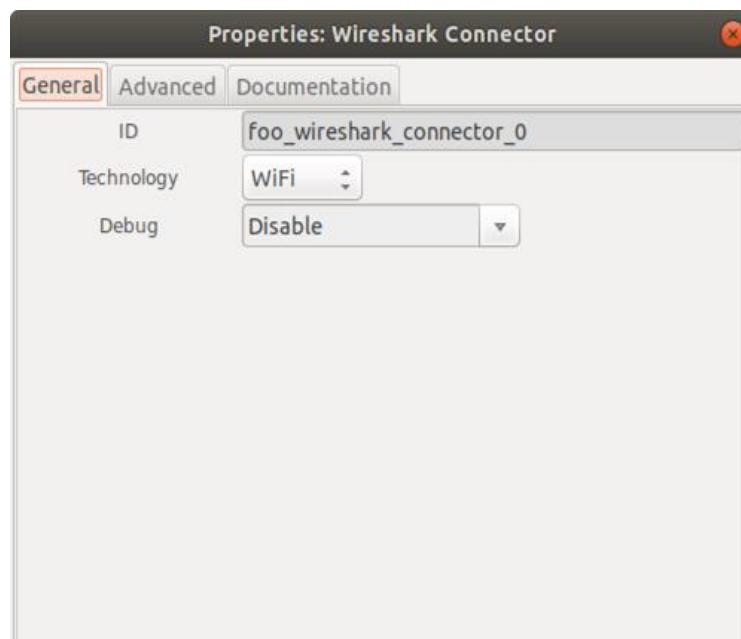


Figura 144 Conector Wireshark

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Bloque File Sink

Sincroniza un archivo

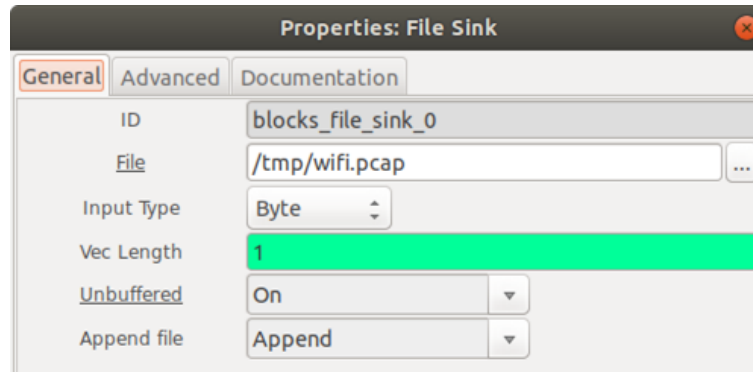
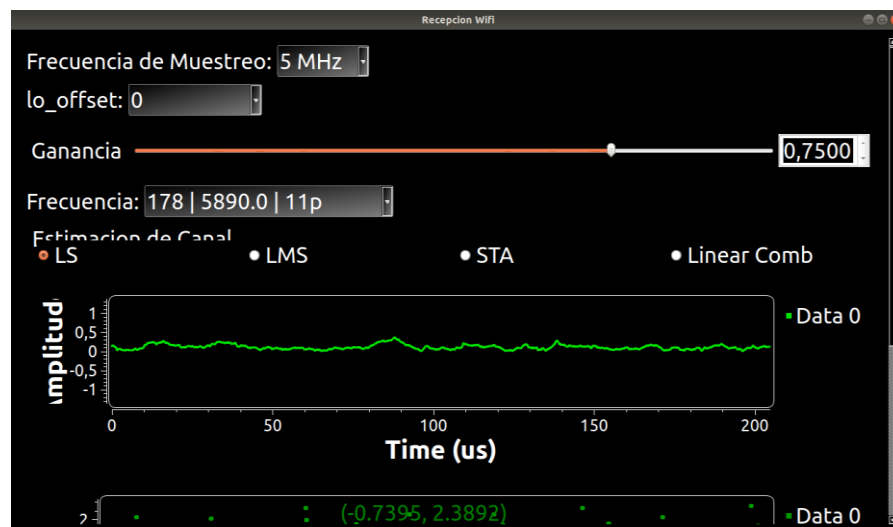


Figura 145 Bloque File Sink

Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

Paso 3

Ejecutamos nuestra aplicación



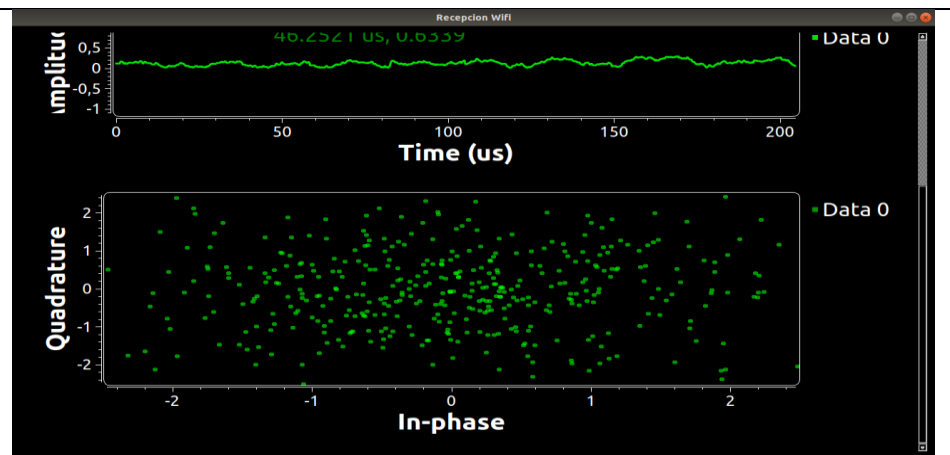


Figura 146 y 147 Paso 3: Ejecución de aplicación
Fuente: Víctor Alejandro Pinza Castro y Jaime Gabriel Pilamunga Morocho

CONCLUSIONES

- ✓ En esta práctica se demuestra el desarrollo de la radio definida por software (SDR) que utiliza el banco de pruebas SDR04 y el kit de herramientas complementario de GNU Radio. IEEE802.11a OFDM se ejecuta con codificación de canal considerando el codificador FEC. IEEE 802.11a, determinaciones de transceptor OFDM en diferentes esquemas de modulación, niveles de ruido del canal y variación de fase.
- ✓ En el estudio comparativo, se encontró que: cuando las condiciones del canal son deficientes, se usaron esquemas de eficiencia energética como BPSK y, a medida que la calidad del canal mejora, se usó QPSK.

RECOMENDACIONES

- ✓ Ajustar el método de modulación casi directamente para una transferencia de datos óptima, lo cual produce mayor eficiencia del ancho de banda y aumenta la capacidad general del sistema.

ACTA DE ENTREGA – RECEPCIÓN DE TESIS.

Jaime Pilamunga Morocho CI: 0924754419 y Víctor Pinza Castro CI: 0923879415, Estudiantes de la carrera de Ingeniería en Electrónica entregamos a la Universidad Politécnica Salesiana el equipo que forma parte del proyecto de titulación previa a la obtención del Título de Ingeniero Electrónico.

Tema:

“DISEÑO DE ALGORITMOS PARA LA SINCRONIZACIÓN Y CODIFICACIÓN DE CANALES DE SISTEMAS OFDM UTILIZANDO RADIOS DEFINIDOS POR SOFTWARE”.

ITEM	NOMBRE-APELLIDOS	Nº CELULAR	CORREO
1	Jaime Pilamunga Morocho	0963126949	jpilamunga@est.ups.edu.ec
2	Víctor Pinza Castro	0994939283	vpinza@est.ups.edu.ec

Descripción del equipo

ITEM #1 USRP B205 mini-i	SE DETALLAN LAS ESPECIFICACIONES TÉCNICAS SECTORIZADAS O POR ETAPAS DE LA TESIS
Detalle Fotográfico Sectorizado	Especificaciones Técnicas
	<ul style="list-style-type: none">• Conector USB 3.0 cable

Tesistas:

Jaime Gabriel Pilamunga Morocho
C.I. 0924754419

Víctor Alejandro Pinza Castro
C.I. 0923879415